2017

Natural language processing of online propaganda as a means of passively monitoring an adversarial ideology

Holm, Raven R.

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/52993

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

NATURAL LANGUAGE PROCESSING OF ONLINE PROPAGANDA AS A MEANS OF PASSIVELY MONITORING AN ADVERSARIAL IDEOLOGY

by

Raven R. Holm

March 2017

Thesis Co-Advisors: Mathias Kölsch
Justin Jones

Approved for public release. Distribution is unlimited.

Reissued 30 May 2017 with Second Reader's non-NPS affiliation added to title page.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave Blank)*	**2. REPORT DATE** March 2017	**3. REPORT TYPE AND DATES COVERED** Master's Thesis 07-05-2016 to 03-31-2017

4. TITLE AND SUBTITLE

NATURAL LANGUAGE PROCESSING OF ONLINE PROPAGANDA AS A MEANS OF PASSIVELY MONITORING AN ADVERSARIAL IDEOLOGY

5. FUNDING NUMBERS

6. AUTHOR(S)

Raven R. Holm

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Online propaganda embodies a potent new form of warfare; one that extends the strategic reach of our adversaries and overwhelms analysts. Foreign organizations have effectively leveraged an online presence to influence elections and distance-recruit. The Islamic State has also shown proficiency in outsourcing violence, proving that propaganda can enable an organization to wage physical war at very little cost and without the resources traditionally required. To augment new counter foreign propaganda initiatives, this thesis presents a pipeline for defining, detecting and monitoring ideology in text. A corpus of 3,049 modern online texts was assembled and two classifiers were created: one for detecting authorship and another for detecting ideology. The classifiers demonstrated 92.70% recall and 95.84% precision in detecting authorship, and detected ideological content with 76.53% recall and 95.61% precision. Both classifiers were combined to simulate how an ideology can be detected and how its composition could be passively monitored across time. Implementation of such a system could conserve manpower in the intelligence community and add a new dimension to analysis. Although this pipeline makes presumptions about the quality and integrity of input, it is a novel contribution to the fields of Natural Language Processing and Information Warfare.

14. SUBJECT TERMS

data mining, natural language processing, machine learning, algorithm design, information warfare, propaganda

15. NUMBER OF PAGES 87

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

NATURAL LANGUAGE PROCESSING OF ONLINE PROPAGANDA AS A MEANS OF PASSIVELY MONITORING AN ADVERSARIAL IDEOLOGY

Raven R. Holm
Lieutenant, United States Coast Guard
B.S., United States Coast Guard Academy, 2011

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2017

Approved by: Mathias Kölsch
Thesis Co-Advisor

Justin Jones, United States Marine Corps (ret)
Thesis Co-Advisor

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Online propaganda embodies a potent new form of warfare; one that extends the strategic reach of our adversaries and overwhelms analysts. Foreign organizations have effectively leveraged an online presence to influence elections and distance-recruit. The Islamic State has also shown proficiency in outsourcing violence, proving that propaganda can enable an organization to wage physical war at very little cost and without the resources traditionally required. To augment new counter foreign propaganda initiatives, this thesis presents a pipeline for defining, detecting and monitoring ideology in text. A corpus of 3,049 modern online texts was assembled and two classifiers were created: one for detecting authorship and another for detecting ideology. The classifiers demonstrated 92.70% recall and 95.84% precision in detecting authorship, and detected ideological content with 76.53% recall and 95.61% precision. Both classifiers were combined to simulate how an ideology can be detected and how its composition could be passively monitored across time. Implementation of such a system could conserve manpower in the intelligence community and add a new dimension to analysis. Although this pipeline makes presumptions about the quality and integrity of input, it is a novel contribution to the fields of Natural Language Processing and Information Warfare.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

API	Application Program Interface
HTML	Hyper Text Markup Language
IDF	Inverse Document Frequency
IS	Islamic State
IW	Information Warfare
KB	Knowledge-Based
LDA	Latent Dirichlet Allocation
NLP	Natural Language Processing
NLTK	Natural Language Tool-Kit
PDF	Portable Document Format
POS	Part of Speech
ROC	Receiver Operating Characteristic
SRG	Semantic Relationship Graph
SVM	Support Vector Machine
TF	Term Frequency
WSD	Word Sense Disambiguation

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

I would like to express my gratitude to Mathias Kölsch and Justin Jones.

Mathias Kölsch gave this work invaluable direction: he pushed me to pursue subjects I did not know existed, and revisit areas where I was weak. Also, many students credit their advisors for ensuring they finished their theses on time; however, for me, finishing a project is akin to confronting my own mortality. So to say Mathias deserves credit for project completion is an understatement. Thank you for your incredible patience and support.

Justin Jones provided incredible context and motivation. His operational experience, anecdotes, and insight into practical machine learning applications kept this thesis relevant and cohesive. Thank you for all your time and encouragement.

I would also like to thank Gail Thomas, Sally Baho and Neil Rowe for their early insights, as well as Craig Whiteside, Lyn Whitaker and former student James Friedlein for all their work in compiling and lending a corpus of Islamic State press releases. These press releases enabled me to test the time-series efficacy of my classifiers--an integral feature of this pipeline.

Lastly, many thanks to Justin Downs and Taylor Paul for their camaraderie and input.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

The amount of propaganda distributed by Islamic State leadership and its sympathizers has exceeded the processing resources of Western counter-terrorism intelligence analysts [1], [2]. Volume of online material continues to grow at a rate of 74% annually [3]. Although overall violence fluctuates, terrorist attacks beyond the Middle East have steadily increased [4]. While violence abroad may be a function of a growing organization facing increasing military pressure at home, it may be more indicative of a long-term strategy to outsource terrorism throughout the world. The Islamic State has already proven particularly adept at recruiting Westerners. In 2015, the UK's *Telegraph* reported that almost 20% of IS fighters were Western European residents or nationals [5].

In any case, the Islamic State's online distance-recruitment initiative is growing in scope and value to its mission. Quality of online publications has increased commensurately. Media strategy has shifted from blunt press releases to professional-grade magazines, as seen in Figure 1.1. These magazines show acute awareness: they include direct responses to remarks made by Western politicians, select teachings from the Quran and Hadith, and even sections tailored to women. However, the same progressiveness that has advanced their propaganda campaign may, if properly leveraged, yield an actionable glimpse into their information warfare strategy

DABIQ
13 ISSUE
1437 RABI' AL-AKHIR
THE RĀFIDAH
FROM IBN SABA' TO THE DAJJĀL
SPRING 1431 | 2011 | ISSUE 5
INSPIRE
THE TSUNAMI OF CHANGE
The unfolding revolution has brought with it a wave of change. Shaykh Anwar explains.
Special: Interview with Shaykh Abu Huraira | Zawahiri: The short & long-term plans after protests
RUMIYAH
ISSUE 1
Foreword
Articles
News

Figure 1.1. Modern Salafi-Jihadist Propaganda. Adapted from: [6].

Magazine covers demonstrating a typical level of quality and professionalism.

1.1 Motivation

Currently, analysis of online propaganda is performed by manual inspection. This is problematic, since human processing capabilities are dwarfed by the volume of online text. It is also problematic, when considered the high personnel turnover rates in the military: with each promotion or job transfer, familiarity is lost, and new operators do not have the time to regain lost experience. Additional factors include biases, fatigue, and the inherent difficulty of assessing the quality of human interpretations.

While this thesis does not seek to replace intelligence operators, it does seek to help conserve manpower by expediting detection. This thesis also hopes to augment human analysis by using machine learning to uncover patterns in data that humans might otherwise miss.

1.2 Area of Research

Research focused on natural language processing (NLP) and machine learning.

Two classifiers were created: one to detect authorship, and another to detect the presence of an ideology. Salafi-jihadism, the ideology of the Islamic State and al Qaeda, was the subject of this work.

To ensure the classifier would be able to detect particularly Salafi-jihadist textual features, a large quantity of similar texts was required. This included peaceful religious texts such as khutbahs (Islamic sermons), and persuasive writings such as opinion-editorials from *The New York Times*. News articles and blogs dating between 2003 and 2016 were also collected, to ensure similar range of topics in both positive and negative corpora, as well as a large range of writing styles representative of the internet at large. In total, 3,049 samples (2,787,579 words) were collected.

This thesis focused on English text. Since the Islamic State and its sympathizers are prolific publishers in many languages, comprehensive studies are possible using just English.

Features were mined from raw text by employing NLP techniques. Features were fed as input to a machine learning classifier, with the goals of attributing authorship with reasonable certainty, extracting meaningful textual features (e.g. concepts), detecting semantic anomalies, and mapping the proliferation of ideologies across time with the ultimate goal of assisting American intelligence efforts.

Finally, a support vector machine was used to contrast methods, then tuned to optimize results. The best-performing pipelines from both authorship and content detection were combined to optimize recall over time.

Secondary to the computer science aspects of this work, the online media campaign of jihadist terrorists was referenced. True to their slogan of “Enduring and Expanding,” what is here referred to as the Islamic State (IS) has undergone many changes in leadership, mission, approach and organization (Tawhid wal-Jihad, al Qaeda, ISI, ISIS, ISIL, IS) since its first attributed act of terrorism in 2002 [4]. This makes its propaganda a potential source of semantic evolution. That being said, the history and inner workings of the Islamic State are far from the technical objectives of this thesis. Rather, this thesis is tailored to the Salafi-jihadist ideology, and will include works by both al Qaeda and IS in the positive corpus without delving into their differences.

1.3 Research Questions

1. How well can Natural Language Processing (NLP) be used to classify online text as propaganda from a particular ideology?
    - With what accuracy can NLP identify propaganda from a particular ideology?
    - What is the data-driven definition of a particular ideology?
2. Can NLP be used to monitor the spread and evolution of an ideology? Spread meaning tracking the prevalence with respect to time and limited in scope by the type of online media, e.g. blogs, vice articles.
    - Can a divergence, trends or anomalies in an ideology be detected?

1.4 Limitations

The biggest limitation on the scope of this thesis was time. It would have been beneficial to test this pipeline on more than one ideology, or to contrast the semantic content of two opposing ideologies, such as the Obama Administration and the Kremlin.

A constraint that this thesis overcame was the scarcity of free and modern online text data, suited to the needs of this thesis. Web scraping met this need, and allowed us to tailor the data to the thesis, however it came at a time cost. And while scraping modern web content took time, it was not nearly as costly as cleaning. Converting online magazines from portable document format (PDF) to plain text, was particularly time-consuming.

This classification system is not intended to establish ground truth. In many cases, authorship and intention may never be conclusively identified. However, enabling users to ascertain with reasonable certainty that texts contain content of interest may increase the volume of data an operator can process with enough success to outweigh the loss of unidentified material.

1.5 Thesis Organization

The following chapters will outline the process used to collect texts, extract meaningful textual features, apply classifiers, integrate learning techniques, and analyze results. The rest of this thesis is organized as follows:

Chapter 2: Background. A discussion of relevant topics, substantiated by prior works.

Chapter 3: Authorship Detection. A thorough description of techniques.

Chapter 4: Content Detection. Three concept extraction methods are contrasted.

Chapter 5: The Combined Pipeline. Authorship and content detectors are paired and applied to time-series data.

Chapter 6: Conclusion. Conclusions and recommendations for future work.

CHAPTER 2:
Background

The following is an overview of relevant topics and prior works. This chapter is intentionally organized to lay the logical foundation of the proposed pipeline

2.1 Data Mining

Data mining is a broad discipline of computer science where new information is acquired from data, analyzed, and applied [7]. Data mining is integral to a number of fields including bioinformatics, NLP, web mining, style mining and opinion mining [8]. Data mining is also referred to as knowledge discovery from data (KDD), a term that better highlights its utility.

Large corpora of cleaned data can be purchased or found for free, or they can be manually compiled, or a combination thereof. While using corpora compiled by someone else has the advantage of a large time-savings, they may not meet the specific needs of the application. Manual compilation has the advantage of being highly tailored to the task; better data lends better results. For this thesis, a corpus was created using a combination of free online data resources, such as opensource.gov, but mostly text manually scraped from the internet.

With petabytes of data being added to the World Wide Web every day, sifting through the internet to find pertinent information is no small task [9]. In the same token, many sectors of industry have acknowledged the vast potential of big data and have leaned forward into the "datafication" of our daily lives: collecting and labeling more data, and developing products such as Application Program Interfaces (API) that allow limited access to large databases [8, p.115]. This grant coders and third-party applications access to valuable data, while also protecting the company's best interests. Programmatically iterating through the web (crawling or spidering) is a common technique for expeditiously locating data that meets certain criteria.

Data mining can also provide a framework for extracting information from text. Data mining tasks can be categorized as either descriptive or predictive [9]. Descriptive tasks are concerned with characterizing data. Predictive tasks make projections about future data after performing induction on old data [9]. Examples of data mining tasks include discrimination, pattern finding, correlation, clustering and anomaly detection.

Previously at NPS, data mining of Islamic State press releases was been performed to classify the documents as one of seven classes: Admin, Attack, Celebrate, Defense, Eulogy, Recruitment, or Strategic Communication [10]. Complimented with geotags and timestamps, this "rich" data was extracted with an interactive tool in mind, to help users visualize trends [10]. This would be an example of a descriptive task.

A predictive task might take the same data, and predict the class of the next press release, or even attempt to generate the text (wording, date, geotag) itself.

2.2 Preprocessing

The phrase "garbage in, garbage out" captures the notion that noisy, unrealistically biased, or low-quality data will result in a poor classifier. However high-quality data which is not handed to the classifier in good shape can also lead to "garbage out." Preprocessing entails the steps it takes to ensure data is free of features that will mislead the classifier.

There are several approaches and levels of preprocessing text. On the individual document level, preprocessing involves cleaning the text of things that could mislead the classifier. This commonly includes stripping the text of punctuation, "stop words," macro data, or extraneous formatting code.

Stop words are terms considered uninformative, and possibly distracting, to the task. Common words, such as prepositions and conjunctions, are typically included in stop word lists since they contribute very little to semantic meaning but their high-usage could skew frequency-based weighting schemes. Stop word lists must be employed carefully, because although they reduce the chances of accumulating irrelevant data, useful information may be discarded [11]. Additionally, the relative frequency of common words can be helpful for certain tasks, such as detecting authorship.

Macro data that is commonly stripped from text includes authors, dates, and source (e.g. Associated Press). Unremoved, macro data could teach a classifier to place too much weight on unreliable features, or it could contain ground truth and unintentionally lead the classifier to “cheat.” For example, in guessing the gender of an author, it would be disadvantageous for a classifier to learn on texts which included names, since it may focus on learning which names are female, instead of learning semantic characteristics. Such a classifier would likely fail when presented with unusual or gender-ambiguous names, like Leslie.

Similarly, when scraping text from the internet, several layers of formatting must be navigated. This includes Extensible Markup Language (XML), Hyper Text Markup Language (HTML) and API-specific formatting. The process of removing these encapsulating languages for a clean text, which reflects what the reader of a web page would see.

Once the visible text has been properly extracted and cleaned, the objective of preprocessing shifts to feature selection.

Definition 2.1 *Features* are machine-detectable attributes of samples, relevant to a machine learning task on the corpus containing the samples.

Once chosen, these features are translated to vector format for machine learning. One common technique for doing this is the Full Text Approach, wherein each document is treated as a “bag of words.” Structural aspects, such as spacing, punctuation, and non-alphabetic characters are removed. This approach retains the “richness” of text; however word-ordering is lost and lots of unhelpful information is retained [11]. The resulting vector would have an element representative of each term in the text, with a value denoting the count. More approaches and features types will be detailed in the next section.

The final step of preprocessing is formatting the feature vector itself. This may involve normalizing, scaling or transforming the data.

2.3 Natural Language Processing

Natural Language Processing (NLP) is the use of computers to quantify features of human language. In this thesis, NLP will be applied to written texts to extract data relevant to passive monitoring of ideological groups. This includes authorship detection, text classification, ontology construction, and anomaly detection. NLP features can be lexical, syntactical or semantic.

2.3.1 Lexical Features

Lexical features include size of vocabulary, term frequency (TF), n-grams, and Part of Speech (POS) tags.

N-grams are counts of adjacent terms. Using the idiom "Between a rock and a hard place" as an example, One-grams would be all of the individual words of the phrase. Bigrams would include "Between a", "a rock", "rock and", "and a", etc. Trigrams would include "Between a rock", "a rock and", and so forth.

Parts of Speech refer to the syntactic function of a given term, such as noun, pronoun, adjective or verb. In line with most NLP applications, this thesis utilized the University of Pennsylvania POS tagging scheme, as shown in Table 2.1.

Table 2.1. UPenn Treebank Part-of-Speech Tags. Adapted from [12].

CC	Coordinating conjunction	PRP$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign Word	RP	Particle
IN	Preposition or subord. conj.	SYM	Symbol
JJ	Adjective	TO	to
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present particle
NN	Noun, singular or mass	VBN	Verb, past participle
NNP	Proper noun, singular	VBP	Verb, non-3rd person singular present
NNPS	Proper noun, plural	VBZ	Verb, 3rd person singular present
NNS	Noun, plural	WDT	Wh-determiner
PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP$	Possessive wh-pronoun
PRP	Personal pronoun	WRB	Wh-adverb

The University of Pennsylvania's Treebank Part-of-Speech (POS) tagging system is implemented in Python's NLTK module. The NLTK module also contains separate tags for: opening quotations, closing quotations, opening parentheses, closing parentheses, commas, dashes, dollar symbols, sentence terminators, dashes, and colons or ellipses.

Parts of Speech counts were integral to authorship detection, as they are excellent indicators of personal writing preferences. An example of an authorship bias detectable with POS counts: a tendency to use three times as many adjectives as nouns.

2.3.2 Syntactical Features

Syntactic features are those concerning the arrangement of words. Examples include grammar habits, sentence structure and POS ordering, as already described. Syntactic features can be thought of as the rules of a given grammar.

Syntax can also be an indicator of native tongue. For example, in Arabic there are two types of sentences: verbal sentences and nominal sentences, which typically follow Verb-Subject-Object and Subject-Predicate word ordering, respectively [13]. This knowledge inspired normalized POS counts in the hopes that it would help differentiate English Second Language (ESL) or translated texts, from those authored by native English speakers, who tend to prefer Subject-Verb-Object ordering. While this thesis focused on English texts, some other subtle POS idiosyncrasies may carry over from a language as morphologically rich as Arabic.

2.3.3 Semantic Features

Semantic features strive to capture meaning, which can be either explicit or abstract. Explicit semantic features, are terms themselves. Theses explicit semantic features were a large component of the authorship detection vector, where rarely-used terms and terms appearing in unusual frequency were key.

Semantic features were also relevant to ideology detection. One such feature is word sense, or the way in which a word is used. For example, in many Middle Eastern cultures the word jihad is not negative. However, when paired with words such as “fight” or “kuffar” (a derogatory name for non-believers), it was deemed indicative of Salafi-jihadist ideology. Through clustering or matrix-reduction, higher-level semantic features can be extracted from texts, without explicit appearance.

These approaches can likewise help distinguish texts of similar topics but opposing stances, commonly referred to as *sentiment* analysis. For example, an opinion piece on the American military has a negative connotation when paired with “bureaucracy,” “assault” and “overspending”; and a positive connotation when paired with “best,” “partnerships” and “professional.”

2.4 Machine Learning

Machine Learning is a field of computer science that gives computers the ability to learn and act without explicit direction. It is particularly adept to big data applications and complex system resolution. In general, a classifier is trained on a model, then it classifies test data based on patterns it learned from the training data.

Machine learning can be supervised, unsupervised, or semi-supervised. In supervised learning, all data including training data, is labeled. Unsupervised learning occurs when data is fed in, and the output is a set of inferences about the hidden structure of data. Clustering is the token example of unsupervised learning.

This thesis performed authorship detection with data labeled as either positive or negative, and was thus supervised. In content detection, this thesis experimented with clustering and matrix resolution techniques to extract rich data from training data (unsupervised). This rich data provided a framework for quantifying labeled training data, for machine learning input. Since content detection combined unsupervised and supervised learning, it is semi-supervised.

Popular forms of machine learning include:

- Naive Bayes
- Clustering
- Support Vector Machines
- Neural Networks
- Ensemble Learning

2.4.1 Naive Bayes

Naive Bayes is a conditional probability model in which a certain class is assigned a probability of occurring, given a certain feature. This probability can be deduced using the prior probability of the class, the likelihood that the feature occurs given the class, and the probability of the feature itself.

$$P(C|f) = \frac{P(C) * P(f|C)}{P(f)}, \text{ where C is a given class, and f a given feature} \tag{2.1}$$

This approach gets its name from the assumption that every feature is naively assumed to be independent from other features, such that the joint probability distribution of a class can be described as a function of a feature vector as follows:

$$P(C|f_1, f_2, ..., f_n) \propto P(C) * \sum_{i=1}^{n} P(f_i|C) \tag{2.2}$$

Naive Bayes classifiers are popular for text classification, wherein a single text is commonly considered a distribution of words [14]. As such, each element in a vector of term counts can be related to a probability corresponding to a class. For example, if the term *extremist* appears in 4% of world news articles, 7% of Opinion pieces published by *The New York Times*, but 78% of magazines, it can be an important classification clue.

Additionally, in Chapter 4, a method called Latent Dirichlet Allocation (LDA) is used to model discrete data describing a corpus. LDA does this with a three-level hierarchical Bayesian model which assumes the corpus is an infinite mixture of extractable topics [15]. This Bayesian approach is used to reduce the dimensionality of term frequency counts: distilling the data into its most potent form. For more on the implementation, see section 4.3.3.

2.4.2 Clustering

Cluster Analysis, or clustering, divides data into groups and is a form of abstraction. These groups are often referred to as classes, and should either have conceptual meaning or utility to their organization [16]. Clustering for conceptual meaning would include the phyla created by scientists studying species, or the genres created for organizing music. Examples of clustering for utility may include computing distances for Recommender Systems, such as the proximity between songs for Pandora suggestions.

Popular clustering techniques include k-means, hierarchical, and DBSCAN. In k-means, k centroids of the data are determined, and data points are grouped with the closest centroid. K-means is popular for its simplicity and efficiency, but does not always work well on non-globular or clusters of different sizes and densities [16].

In hierarchical (also referred to as “agglomerative”) clustering, each data point is initially considered its own cluster. With each new iteration, clusters are merged with the next closest cluster, or as based on a increasing distance threshold. This normally results in a tree-like structure, called a dendogram. The resulting dendogram typically motivates the usage of hierarchical clustering: the user is generally seeking to create a taxonomy or phylum. However hierarchical clustering has also been shown to perform well against other clustering approaches for purposes other than taxonomy creation [16]. On the other hand, it is an expensive and sometimes noisy process that may require manual fine tuning.

Density Based Spatial Clustering of Applications with Noise (DBSCAN) originates core clusters in highest-density areas, then expands outwards. DBSCAN is well-suited for analyzing data of visually obvious clusters, or clusters of similar density.

In the past, clustering has been used to cluster and classify documents found in the Arabic NEWSWIRE corpus [17]. Since it is so morphologically rich, Arabic evades many rule-based approaches that work on Germanic and Romance languages [17]. As a form of data abstraction in itself, clustering has been shown to be more resilient to morphology [17].

Similarly, clustering can be used to group documents of similar subject matter to arrive at higher level abstractions. Inspection of machine-learned clusters can then reveal lower-level discriminative features For example, the context of an otherwise innocuous phrase which may support a topic associated with Salafi-jihadist ideology. For more about the implementation of clustering, see section 4.3.2.

2.4.3 Support Vector Machines

Support Vector Machines (SVM) are classification algorithms that learn hyperplanes to best delineate high-dimensional feature spaces [18]. Different decision surfaces are commonly available, notably including linear, polynomial and exponential functions. When applied to learning models, these functions are often referred to as kernel functions. Classifier hyperparameters are tuned to include the most data possible without overfitting. Kernel functions are traditionally depicted as mapping a nonlinear decision space into a neatly plane-delineated feature space, as shown in Figure 2.1.

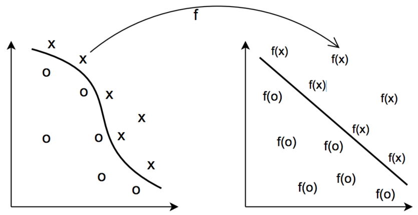


Figure 2.1. Visualization of a Kernel Function. Adapted from: [19].

SVMs map positive and negative data to a space where they can be delineated by a boundary. This decision space becomes the basis for classification of new data. This is also referred to as the "kernel trick."

Compared to neural networks, SVM's are easy to train and have fewer parameters. Since so much of this work's time allotment was consumed by assembling data and finding the best method and combination of features, SVM's were chosen as the classification means.

2.4.4 Neural Networks

Artificial neural networks, referred to here as neural networks for brevity, take their inspiration from the natural neural networks of the brain. Human brains are comprised of approximately 10^9 neurons, connected by synapses and operating in parallel [20]. By simply associating inputs with outputs, but in very large scale, neural networks can resolve extremely complex pattern spaces. The many architecture and fine-tuning possibilities make neural networks very adaptive to different problem spaces.

Neural networks have been applied effectively to NLP applications, such as sentiment analysis on short text [21]. That being said, neural networks do not always offer much in the way of explanation as to how they arrive at their conclusions. Rule-extraction algorithms are steadily improving the transparency of networks; mathematical expressions, logic statements, and even decision trees can now be derived from neural networks [22]. However, applying and making sense of these extracted-rules would likely require a level of expertise in neural networks as well as negate the time-savings this thesis strove for. So for this reason, in addition to those stated in Section 2.4.3, neural networks were not used in this work.

2.4.5 Ensemble Learning

In ensemble learning, various machine learning techniques are combined to make a single classification decision. By optimally combining and factoring the lower level learning techniques, the combined classifier can mitigate weaknesses of each technique and arrive at accuracy higher than that of its individual learning components. Ensemble learning will be used to combine multiple machine learning methods in Chapter 6, with the goal of maximizing recall for intelligence applications.

2.5 Evaluation Techniques

There are several popular classifier evaluation metrics:

Precision: the number of accurately classified objects (true positives), out of all objects which should have been classified (both true positives and false positives).

$$Precision = \frac{tp}{tp + fp} \tag{2.3}$$

Recall: the number of accurately classified objects, out of all objects classified as positive (both true positives and false negatives).

$$Recall = \frac{tp}{tp + fn} \tag{2.4}$$

F-score: the harmonic mean of Precision and Recall. Also referred to as the F1-Score and F-measure.

$$F - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.5}$$

Accuracy: the proportion of true results to total results.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{2.6}$$

Receiver Operating Characteristic (ROC) Curve: graphs showing the performance of a classifier by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR).

Chapters 3 and 4 of this thesis focused on accuracy of their respective classifiers. Chapter 5 prioritized recall. In all chapters, ROC curves were the primary tool for performance visualization.

Many additional techniques exist to further investigate and validate classifier performance. Two such techniques which were made use of:

Best-Feature Extraction: In best-feature extraction, the evaluator is able to manually verify the features that the classifier weights highest. For example, if a high-scoring classifier is intended to detect gender in blogs, but focuses on the terms “the,” “is” and “12,” it may prompt the evaluator to either employ a stop word list or get a larger corpus.

Cross-Validation: In cross-validation, training and testing is run multiple times, each time on distinctly separate data. This ensures that classifier results are not coincidentally good in a single test, or that the classifier is overfitting to training data and will fail on new data.

Both techniques were available as built-in Python functions.

2.6 Propaganda

In their book *Propaganda and Persuasion* [23], Jowett and O’Donnell consider both propaganda and persuasion to be categories of communication. They define propaganda as “a form of communication that attempts to achieve a response that furthers the desired intent of the propagandist” [23]. They differentiate this from persuasion, which "is interactive and attempts to satisfy the needs of both persuader and persuadee” [23]. So while propaganda may or may not utilize persuasive techniques, it is always characterized by an intent to disseminate particular ideas.

Definition 2.2 In this thesis, *propaganda* will be considered texts with a strong association with an ideology, which may or may not be persuasive.

Definition 2.3 An *ideology* will be defined as a particular set of semantic concepts.

By considering texts as distributions of words, and words as the basis of higher-level semantic features (concepts), it is possible to both define and detect concepts as distributions of words. Thus, it is possible to detect an ideology, and classify a given text as propaganda or not propaganda.

2.7 Ideology

While already defined in NLP terms, ideologies are also worth discussing in anthropoligical terms. In this optic, ideologies are narratives that extol certain beliefs and call for social, economic or political change. They are typically organized manifestations of frustration, fear, or perceived unfairness in the face of modernization, globalization, or crisis [24]. They aim to recruit a particular group of people, while casting blame on another group. They can be religious or secular, but they always meet four core functions [24]:

1. Raise awareness for an issue
2. Attribute blame
3. Promote a group identity
4. Prescribe action

Examples of ideologies include National Socialism (Nazi), Feminism, Paleoconservatism (Right to Life, Prohibition), Libertarianism, and the opposing national narratives of Russia and the United States. One thing that makes ideologies so ubiquitous is that they provide frameworks [24]. Frameworks make it easy to link cause and effect, and simplify social realities; this efficiency makes them self-reinforcing on a subconscious level. Frameworks also hold special appeal for people wishing to assign blame during times of crisis or change [24].

2.7.1 Salafi-jihad

The Salafi-jihad is the doctrine of the Islamic State, its predecessors and its affiliates including al Qaeda. It is considered a hybrid religious ideology, which combines the Salafi movement with a unique interpretation of jihad [24].

The Salafi movement is an ultraconservative branch of Sunni Islam. In the 19th century, it merged with the fundamentalist and highly intolerant Wahhabi branch - also Sunni, and predominantly associated with Saudi Arabia [25]. Salafism and Wahhabism are now considered synonymous. They both evolved in reaction to European colonialism and industrialization, as is characteristic of most ideologies. Salafism advocates Sharia law: an ultraconservative and highly intolerant form of law with selective origins in the Quran, the Hadith, and regional tradition [25].

Although it self-describes as a true interpretation of Islam, the Salafi-jihad is closer to an ideology than a religion, in both its implementation and in that its interpretation of Islam is highly controversial [24]. The Salafi-jihad meets the four defining functions of an ideology: it promotes the image of Islam as under attack by non-Muslims, and Western "crusaders" in particular; it blames Europe for using colonization and diplomacy to prevent its economic rise and relegating its culture; it promotes a superior and intolerant Muslim-soldier identity; and it advocates certain actions, such as adopting sharia and acts of terror [26], [24].

Classifying Salafi-jihadism as a religious ideology is important to combating it: this distinction can help diffuse its objectives of pinning Christianity as against Islam, and unifying Muslims under a radical and combative interpretation. Refusing to acknowledge its religious qualities and focusing on its secular faults, such as violence and corruption, is thought to be a more viable strategy [24]. Yet truly effective countermeasures require a heightened awareness of what the adversary is saying, requiring a system such as the one this thesis proposes.

2.8 Information Warfare and National Strategy

As unconventional warfare in the digital sphere becomes more and more prevalent, the United States continues to expand Information Warfare initiatives. In December of 2016, the Countering Foreign Propaganda and Disinformation Act was passed as part of Fiscal Year 2017's National Defense Authorization [27]. The bill designates $160 million towards identifying and combating propaganda [28].

While the bill was ostensibly an expression of concern over reports of Russian meddling in the 2016 presidential election, it broadly targets all foreign disinformation and propaganda, and even names the People's Republic of China [27]. The Obama administration also announced plans to create a counterterrorism task force to specifically combat the flood of online propaganda by terrorist organizations [29]. In many ways, Information Warfare may be the most effective way to wage war against non-state actors posing a threat to our Nation.

In *Information Strategy and Warfare*, John Arquilla opines that while al Qaeda is extremely competent at information warfare, its core strength lies in its ability to execute a "war of ideas" [30]. This can manifest in many ways, from symbolic messaging, to psychological operations.

2.8.1 Symbolism

For a religious terrorist organization such as the Islamic State, understanding the symbolism contained in their messages is of particular importance, since it often holds operational or tactical significance. Osama bin Laden planned the 9/11 attacks as a symbolic return of Islam, in commemoration of the battle on September 11, 1683, in which the Kind of Poland defeated the Muslim armies [31]. Similarly, the first bombing of the World Trade Center was originally scheduled for February 23, 1993 - in response to the first day of the American ground offensive in Iraq, 1991 [31].

2.8.2 Psychological Operations (PSYOP)

Psychological Operations are operations, normally involving propaganda, which seek to evoke a psychological response. This can include fostering hatred, or creating unity. PSYOPs are a logical extension of ideologies, which typically seek to assign blame to one group, while promoting another (see section 2.7).

Thus, it is easy to see how an adversarial ideology could translate to the four lines of effort commonly used to attack organizations. Roughly paraphrased here, the four lines of effort are [32]:

1. Defection: Recruit individuals within the organization/population capable of conducting resistance
2. Division: Conduct psychological operations to disrupt unity
3. Deception: Protect the true intentions of the adversary
4. Diversion: Divert or misdirect attention from the true source of efforts

The increased efficiency and quality of analysis this thesis seeks to provide, could be applied to counter both *Deception* and *Division* lines of effort, in the following ways:

- A better understanding of the messaging an organization uses to encourage defection, could be used to discourage it on a legitimate and public platform. Example: President Obama rejected the term "Radical Islam" to undermine IS efforts to unify Islam against Christianity.
- Information to erode trust or delegitimize an organizations or media arms could be disseminated. For example, ISI's Ministry of Information goes to great lengths to legitimize itself. It could prove a relatively inexpensive and easy target for a delegitimization campaign.
- Information pointing out hypocrisy in organizational messages could be disseminated. This is a common IS tactic: if it is effective for recruiting, it may be just as effective for disrupting recruitment.
- Identify the core of what an organization is espousing, and conduct retaliation operations specific to their messaging. Example: Identifying negative sentiment from the Kremlin on certain subjects could guide an Information Warfare response to the manipulation of an American presidential election.

In his book *Information Warfare and Organizational Decision-Making*, Alexander Kott argues that the very purpose of an organization is to process more information than a given individual, with the goal of increased decision quality [32]. The broader implications of a system that can efficiently identify adversarial ideological content, is improved Information Warfare decision making, and a better executed National Strategy.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Authorship

3.1 Overview

This chapter outlines and evaluates a method for detecting texts of a given author. It could be applied to locating texts from a known propagandist, or attributing authorship across a set of texts expressing the same ideology.

Identifiers of authorship are primarily syntactical and lexicographical [33]. Examples include average sentence length and vocabulary, respectively.

It was originally thought that available publications of Salafi-jihadists would be too few for machine learning techniques to be applied without overfitting, and likely too difficult to attribute authorship labels with certainty. Thus, this method was first vetted with articles from the Health Section of *The Atlantic*. This dataset covers similar subject matter, contains a persuasive element, and shares a common publisher and editor. These unifying elements were sought to mimic the cohesion of a propagandist organization. Additionally, *The Atlantic* does not require a membership to access its pages, which made it an excellent candidate for automated web scraping.

3.2 Methodology

This section outlines how samples were translated into feature vectors, preprocessing techniques, and machine learning parameters.

3.2.1 Features and Definitions

Features for detecting authorship:

- Lexical Diversity
- Average Characters Per Word
- Standard Deviation of Characters Per Word
- Average Words Per Sentence

- Standard Deviation of Words Per Sentence
- Parts of Speech Counts, normalized across sample length
- Punctuation Counts, normalized across sample length
- Term-Frequency Inverse-Document-Frequency

Word Count is defined as the total number of words appearing in the document. “Word” and “term” are used interchangeably, as are “document” and “sample.” WC became the normalization factor for other features, and did not make it to the final feature vector.

$$WC = \Sigma\ terms \in d, \text{where } d \text{ is a given document} \tag{3.1}$$

Lexical Diversity is defined as the number of unique terms appearing in a document, divided by the number of terms total.

$$LD = \frac{\Sigma\ terms_{unique} \in d}{WC} \tag{3.2}$$

Average Characters Per Word is calculated by averaging the sum of characters in each term appearing in the sample.

$$CPW_{ave} = \frac{\Sigma\ chars \in term}{\Sigma\ terms \in d} \tag{3.3}$$

Standard Deviation of Characters Per Word is calculated by finding the distribution of Characters Per Word per sample.

$$CPW_{dev} = \sqrt{\frac{\sum(CPW_t - CPW_{ave})^2}{\Sigma terms \in d}}, \text{where } t \text{ is a given term in the document} \tag{3.4}$$

Average Word Per Sentence is calculated by dividing the sum of terms appearing in the sample, by the number of sentences appearing in the sample.

$$WPS_{ave} = \frac{\sum terms \in d}{\sum sentences \in d} \tag{3.5}$$

Standard Deviation of Word Per Sentence is calculated by finding the distribution of Words

Per Sentence appearing in the sample.

$$WPS_{dev} = \sqrt{\frac{\sum(WPS_s - WPS_{ave})^2}{\Sigma sentences \in d}}, \text{ where } s \text{ is a given sentence in the document} \quad (3.6)$$

Part of Speech (POS) counts are calculated using NLTK's built-in Parts of Speech tagging function. For each sample, 36 POS counts were tallied, reflecting one count for each of the possible POS tags (see Table 2.1). These counts were normalized across word count (WC).

This thesis explored the notion of "embeddedness"; that is to say, we were interested in how often a writer leveraged quotations and other possible indicators of persuasive writing, such as ellipses. For that reason, quotes, apostrophes, ellipses, enclosures (e.g. brackets and parenthesis) were also tallied and normalized by word count (WC).

Semicolons, colons, periods, commas, question marks, exclamation marks, dashes, dollar signs and percentage signs were also counted and normalized by word count (WC), as likely indicators of authorship.

Term Frequency (TF) is the frequency with which a term occurs over all terms in the sample. TF is calculated once for each document, for each unique term in the corpus. TF does not become part of the final feature vector, it is only used to calculate TF-IDF.

$$TF(t, d) = \frac{\sum occurrences_{term} \in d}{WC} \quad (3.7)$$

Inverse Document Frequency (IDF) is the logarithmically scaled inverse fraction of the documents which contain the word. It is an indicator of how important a term is to the entire corpus (D). IDF does not become part of the final feature vector, it is only used to calculate TF-IDF.

$$IDF(t, D) = log(\frac{D}{\Sigma d : t \in d}), \text{ where } D \text{ is the total number of documents in the corpus} \quad (3.8)$$

Term Frequency Inverse Document Frequency (TF-IDF) is the product of TF and IDF. It is calculated once per term in the corpus.

$$TF - IDF(t, d, D) = TF(t, d) * IDF(t, D) \tag{3.9}$$

The composition of the final feature vector for authorship detection can be seen in Figure 3.1.

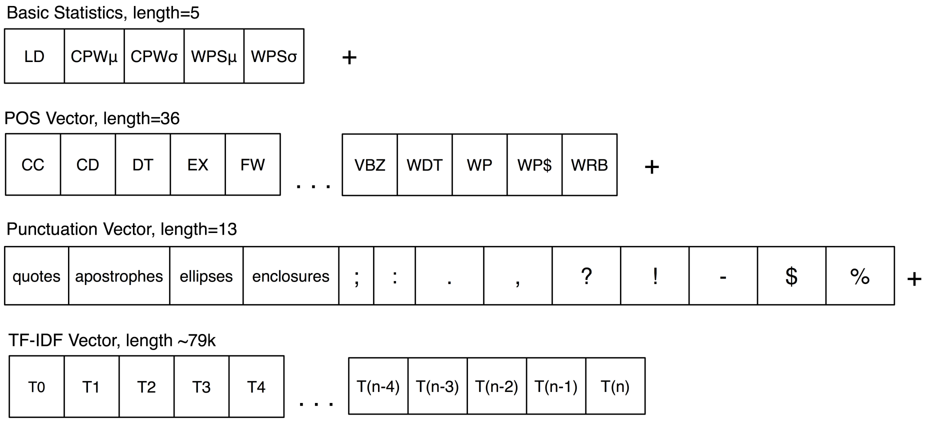


Figure 3.1. Feature Vector for Authorship Detection.

This figure shows the composition of the final feature vector for authorship detection. It is made up of four smaller feature vectors: basic document statistics, parts-of-speech (POS), punctuation and TF-IDF. POS and punctuation counts are each individually normalized by document word count.

3.2.2 Support Vector Machine Parameters

A grid search of Support Vector Machine (SVM) hyperparameters was performed with cross-validation to determine the optimal combination. Namely, type of classifier, C-value, gamma-value and kernel.

C is the regularization variable. A low C makes a smooth decision surface, where a large C aims to correctly classify all values. Too small of a C could mean more false negatives, too large a C could lead to higher accuracy on training, but an overfit model.

Some estimators use alpha as a regularization parameter instead of C [34]. The relationship between the two is given by:

$$C = \frac{D}{\alpha} \tag{3.10}$$

Gamma is a measure of how much influence a single training example has. It inversely affects the radius of influence of the sample, so a high gamma value means a tight radius and a low gamma is a large one [34]. Thus, too low of a gamma can lead to overfitting, and too high of a gamma can prevent the SVM from learning enough.

Types of kernels that were tested:

- Linear
- Polynomial
- Radial Basis Function (default)
- Sigmoid

The Polynomial kernel also has a degree variable. Values 1, 2, 3, 4 and 5 were tested (default is 3).

3.3 Evaluation

This section describes the data and results, and evaluates the approach.

3.3.1 Data

Table 3.1 shows the composition of data used for testing authorship detection.

Table 3.1. Authorship Detection Data.

Dataset	Samples	Authors	Sample Length
Dr. James Hamblin (editor)	446	1	900
Not Dr. James Hamblin	454	39	1,063
All Health Section Authors	900	40	982

This table shows the composition of data scraped from the Health Section of *The Atlantic*. This served as the initial test subject for developing an authorship detector.

Articles were published between March 2015 and September 2016. Each document correlates to one sample; sample length is considered the number of words in each sample.

Since articles were scraped from the internet, web formatting script had to be removed. Sample length is exclusive of non-visible texts.

Articles that were significantly different in size from the target Hamblin-data (greater than 4,000 words, or fewer than 300) were removed. Upon further inspection, unusual sizes typically indicated non-articles. That is to say, very small texts were normally descriptors of embedded media, and very large texts tended to be compilations of other sources.

Additionally, terms that would unfairly bias the machine learning process were removed. Specifically, dates and names of authors were either stripped from the text file or added to a stop word list.

This method assumes input is cleaned (stripped of XML, html, and other non-visible web formatting) and that each document represents one instance of writing. Some web content, e.g. blogs, may have multiple date entries on a given page. Other content may be split across multiple pages, as is common with large texts. It is likely that this method will be less accurate on uncleaned, un-preprocessed data.

3.3.2 Preprocessing

TF-IDF already introduced an element of scaling to the feature vector. To ensure the feature vector as a whole was optimally formatted, the combinations of scaling and normalizing outlined in Table 3.2 were tested on the corpus.

Table 3.2. Scaling/Normalization Method Determination.

Method	Including TF-IDF	Excluding TF-IDF
Just Normalizing	mu = 90.55%, sigma = 3.47%	mu = 86.01%, sigma = 5.49%
Just Scaling	mu = 92.56%, sigma = 3.39%	mu =89.65%, sigma = 3.43%
Both	mu = 92.56%, sigma = 3.39%	mu =88.55%, sigma = 4.63%
Neither	mu =83.90%, sigma = 6.91%	mu =83.90%, sigma = 6.91%

To ensure the best combination of vector formatting was chosen for the authorship pipeline, accuracy of each approach was cross-validated and compared. Since normalization appeared to add no value, scaling became the chosen method (inclusive of TF-IDF).

Results were achieved using cross validation optimized for accuracy, cv=10, and an SVM with kernel set to linear, and C=1.

Since normalizing added no value, future tests were performed with just scaling, but inclusive of the inverse-logarithmically-scaled TF-IDF vector.

3.3.3 Machine Learning

Optimal SVM parameters were determined using a grid search with cross-validation. Tested combinations are outlined in Table 3.3.

Table 3.3. Parameters Tested to Find Optimal SVM.

Kernel	Tested Parameter Values
Linear	C=[1,10,100,100]
Polynomial	C=[1,10,100,100], Gamma=[0.01, 0.001, 0.0001, 0.00001], Degree=[1,2,3,4,5]
RBF	C=[1,10,100,100], Gamma=[0.01, 0.001, 0.0001, 0.00001]
Sigmoid	C=[1,10,100,100], Gamma=[0.01, 0.001, 0.0001, 0.00001]

To find the optimal SVM settings, these parameters were tested.

Using precision as a scoring metric, the combination with the best score was a Sigmoid kernel, C=1, gamma=.0001.

With 10-fold cross-validation, the classifier achieved 88% accuracy. Feature inspection revealed that there was some redundancy in POS tagging and punctuation features. This led us to discover that the NLTK implementation of UPenn's POS tag system had additional tags for punctuation (see Table 2.1). After removing the punctuation features, performance improved to an F-Score of 95.56%, as shown in Table 3.4.

Table 3.4. Authorship Detection Results.

	Precision	Recall	F1-score	Support
0.0	93.02%	97.56%	95.24%	41
1.0	97.87%	93.88%	95.83%	49
avg/total	95.66%	95.56%	95.56%	90

Cross-validated authorship detection results.

Further feature inspection revealed that more terms needed to be added to the stop list, namely "science" and "business." These terms referred to other sections of the magazine, and while the article was also published under the Health Section, these terms were strong indicators of a writer from another section. After these terms were added to the stop list, accuracy dropped to 94.22%. Recall was measured at 92.70%, and precision at 95.84%. ROC curve is shown in Figure 3.2. Feature inspection was performed with scikit-learn's `featureselection.SelectKBest` and `featureselection.VarianceThreshold`.

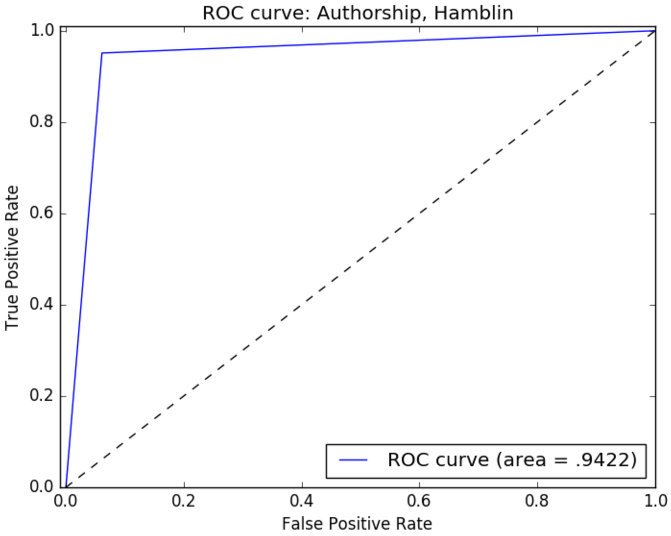


Figure 3.2. ROC Curve for Authorship Detection.

Receiver Operating Characteristic (ROC) curve for detecting authorship. Articles authored by Dr. James Hamblin, for *The Atlantic* were discriminated against articles authored by other Health Section authors.

3.3.4 Validation

Since Dr. James Hamblin is the editor of the entire section, it was thought that the classifier should be retested on another author for validation. Thus, this experiment was repeated using the same features, data, and SVM parameters, but with articles authored by Julie Beck labeled positive, and all others labeled negative (see Table 3.5).

Table 3.5. Authorship Detection Validation Data.

Dataset	Samples	Authors	Ave Sample Length
Julie Beck	125	1	1,042
Not Julie Beck	775	39	979
All Health Section Authors	900	40	982

Authorship detection was re-tested, with an author (Julie Beck) other than the editor (James Hamblin) labeled as the positive case. This table describes the redistributed data.

With cv=10, the classifier performed with 87.11% accuracy and 99.61% recall. The decrease in performance could be attributed to imbalanced classes. That is to say, the SVM develops a bias for the larger (non-Julie Beck) class since it occurs most often.

3.4 Chapter Summary

With cross-validation (cv=10), a SVM (sigmoid kernel, C=1, gamma=.0001) performed with 94.22% accuracy on an author representing approximately 50% of the data, and 87.11% accuracy on an author representing 14% of the data. Recall was recorded at 92.70% and 99.61%, respectively.

Best practices included scaling, inclusive of the TF-IDF vector, and no normalization. Additionally, Parts-of-Speech counts were shown to have a negative effect on the classifier's performance. That being said, POS counts are often an important identifier of original language, and may have a positive effect on data translated from Arabic, or authored by an English Second Language speaker.

This classifier is integrated into the pipeline outlined in Chapter 5.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4: Content Detection

Authorship detection laid the groundwork for detecting text as based on syntactic and lexical features. In this chapter, content detection will be shown to leverage higher-level semantic features for two utilities:

1. To compliment authorship for a more comprehensive detection scheme, or
2. To uncover semantic features for subsequent human analysis and monitoring

4.1 Overview

For this thesis, the goal of content detection is to discriminate between texts containing an ideology, and texts that do not.

The crucial first step for the content detection method presented here was to build high-level features to describe the underlying topics distributed throughout the corpus. These high-level features are referred to interchangeably as “concepts” or “topics,” and were sought using one of three methods:

- Semantic Relationship Graph (SRG) construction
- Clustering
- Latent Dirichlet Allocation (LDA)

Theses features served as the basis for a feature vector, for Support Vector Machine input.

4.2 Definitions

Definition 4.1 *Semantic Relationships* are relationships between terms that can serve as a basis for a *semantic network* . Examples of semantic relationships include synonyms or root words, which can both be acquired with a dictionary.

Definition 4.2 *Ontology* is defined here as a knowledge-based hierarchy of terms. “Knowledge-based” (KB) refers to its construction with pre-established knowledge, in this case: semantic relationships. Working higher in the ontology leads to abstraction; working lower leads to specialization.

Definition 4.3 *Semantic Networks* are networks of terms, connected by relationships. In this work, semantic networks are considered different from ontologies in that they are not necessarily hierarchical. They can also be structured horizontally (e.g. synonyms), or sprawling, complex webs comprised of many relationships. For example, a network of movie titles could be connected by ratings, actors, genre, year of release, etc.

Definition 4.4 *Hyponym/Hypernym* - a *hypernym* is a term found "over" a root term; i.e. the root term has a <type of> relationship with the *hyponym*. For example, "giraffe" is a <type of> "animal." In this case, giraffe is a hyponym and animal is a hypernym.

Definition 4.5 *Topics/Concepts* - *topics*, also referred to as *concepts*, are groupings of terms which describe and help differentiate a corpus. Topics can be uncovered via semantic relationships, cluster analysis, or probabilistic models. Depending on the model, topics may be defined by a keyword, or merely numbered for reference purposes only.

Definition 4.6 `supporting_terms` - the terms which substantiate topics. They have strong associations to a topic within the topic's respective dataset, group or cluster. Depending on the method, they may be accompanied by a factor, which relates their significance to their respective topic. This relationship can be discovered using a probabilistic model or analysis after clustering.

Definition 4.7 `chunk_size` [LDA parameter] - the length of text a topic-extraction algorithm is allowed to work on at a given time. Smaller chunks require less memory, however larger chunks may present more context, which may lead to improved performance.

Definition 4.8 `num_passes` [LDA parameter]- the number of passes the algorithm makes across the corpus. Each pass updates the model, aiding topic convergence. More passes lead to a better model, but require more time.

4.3 Methodologies

This section explains four approaches to uncovering a semantic feature space.

4.3.1 Method1: Semantic Relationship Graph Construction

For SRG construction, concept extraction was guided by the method outlined by Gelfand, Wulfekuhler and Punch in their 1998 paper *Automated Concept Extraction from Plain Text* [35].

This method meets the criteria of what is often referred to as the "linguistic approach": semantic relationships of raw, low-level terms are used to arrive at higher-level concepts [36]. Specifically, this method advocates Princeton's WordNet for uncovering semantic relationships [37].

WordNet is a large database of English terms grouped into 117,000 cognitive synonyms called "synsets." These synsets meet the hypernym-hyponym relationship defined in Section 4.2. WordNet is built into Python's NLTK module, and can be called upon to look up hypernyms of words found in a given text ("base words"). When base words and hyponyms appear with a certain frequency (based on a threshold), they continue on to the next round of WordNet hyponym lookups and so forth, until a highly distilled representation of the text is reached.

The steps of this process, as outlined by Gelfand, Wulfekuhler and Punch, are paraphrased here:

1. Compile an initial list of words ("base words")
2. Perform WordNet lookup of hypernyms and hyponyms
3. Graph parents and sense (see Section 2.3.3) of each hypernym and hyponym looked up
4. Words that link base words ("augmenting words") are added to the Graph, organized by search-level depth
5. Repeat WordNet lookup process on new words, until a predefined search depth is reached
6. Apply thresholds, discarding words that have too few connections to base words
7. What remains are concepts, defined by WordNet-derived semantic relationships. Goal

is to create a structure that connects as many base words as possible

4.3.2 Method2: Clustering with K-Means

Several popular NLP clustering applications posit that clustering documents induces the clustering of terms [38]. A popular technique is to convert each document into a TF or TF-IDF vector, then apply a clustering algorithm [39]. This technique produces clusters of documents, as a function of the distance between cluster centroids and documents in vector space.

In this experiment, each document of the training set was converted into a vector (both TF and TF-IDF were tested), then clustered with K-Means, with K=[6,36,216]. Distances were determined with cosine similarity.

Cosine similarity (S_{cos}) is a measure of the distance between two vectors, defined as follows:

$$S_{cos} = cos(\theta) = cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{y}||} \tag{4.1}$$

Since cosine similarity is a measure of closeness on a scale of zero to one, distance is one minus the cosine similarity, or:

$$dist = 1 - S_{cos} \tag{4.2}$$

After clusters were calculated, a *keyterm* was extracted from each cluster in order of importance, that is to say, distance to the cluster centroid. This was done by determining which vector element, of all vectors in the cluster, most corresponded with the cluster centroid, and relating the element index back to the term it represents.

Definition 4.9 *Keyterm* - the term with the strongest association to a given cluster. This is determined by clustering TF or TF-IDF vector representations of documents, then correlating the vector element closest to the cluster's centroid, with the term it represents.

To improve the power of these cluster-specific keyterms, they were augmented with co-occurring terms, called *supporting_terms*. Co-occurrence was determined by in-cluster bigram analysis. Trigrams and other n-grams were considered, but not tested due to time constraints. The bigrams containing the keyword, were stripped of the keyword, then added to the topic definition.

For example, if the given topic was *president*, bigrams from the News dataset would look like: [‘position’, ‘president’], [‘president’, ‘heinz’], [‘whether’, ‘president’], [‘president’, ‘trump’], [‘hollande’, ‘president’],[‘president’, ‘socialist’], [‘comparable’, ‘president’], [‘president’, ‘obama’], [‘elected’, ‘president’], [‘president’, ‘government’], [‘mexican’, ‘president’], [‘government’, ‘president’], [‘president’, ‘enrique’],[‘government’, ‘president’]... etc

The resulting topic definition: ‘president’:{‘obama’: 234, ‘russian’: 76, ‘vladimir’: 72, ‘bashar’: 71, ‘vice’: 58, ‘recep’: 44, ‘syrian’: 40, ‘former’: 38, ‘turkish’: 38, ‘said’: 35}

These resulting topic definitions give a sense of cluster-specific usage, making it easier to tell if a topic exists without explicit inclusion of the topic term and, conversely, if the topic term is present but in an out-of-cluster context.

For example, the topic definition for *president*, but from the Blogs dataset: ‘president’:{‘obama’: 2, ‘former’: 2, ‘visits’: 1, ‘gerry’: 1, ‘new’: 1, ‘legislation’: 1, ‘moved’: 1, ‘ironically’: 1, ‘made’: 1, ‘spending’: 1}.

Clearly, the term *president* has less significance to the Blogs dataset, even after normalizing for number of documents. The term may also have a less global context: Presidents Recep, Putin and Bashar do not appear to be mentioned in the blogs.

Ideally, the extra terms that make up the topic definitions yield more precise classification than cluster keywords alone. These extra terms are called `supporting_terms`.

Definition 4.10 *supporting_term* - the terms with the strongest association to a given keyterm. This is determined by in-cluster bigram analysis.

Keyterms and `supporting_terms` were complimented with a factor (f_{term}) to relate how important the term was to a given topic. This factor was considered the in-cluster count, divided by the total corpus count, and normalized by the number of documents in the cluster.

$$f_{term} = \frac{tc_C/tc_D}{len(D)}, \qquad where\ C\ is\ a\ topic's\ respective\ cluster,\ and\ D\ is\ the\ corpus \tag{4.3}$$

Note that while converting each document into a feature vector, it is important that the ordering of topics remains consistent. To ensure this, an `OrderedDict` format was used from Python's `collections` module.

A Support Vector Machine was fit to training feature vectors, and fed testing feature vectors for predictions.

The following clustering parameters were tested: TF versus TF-IDF, K=[6,36,216] and `supporting_terms`=[10,20].

To ensure clustering-generated keyterms (e.g. *president*) made it to the final feature vector, they were considered `supporting_terms`. That is to say, when the number of `supporting_terms` was set to 10, one element of the topic feature vector referenced the keyterm, and the other nine corresponded with `supporting_terms`.

4.3.3 Method3: Latent Dirichlet Allocation

LDA is a generative probabilistic model of a corpus. It gets its name from Peter Dirichlet's continuous multivariate probability distribution.

In LDA for NLP, a sample is usually defined as a bag of words representation of a given document. It presumes that each sample is comprised of a mixture of topics, and every term occurring in the document can be attributed to one of these topics [15]. Each topic is also presumed to have a Dirichlet prior probability distribution.

All corpus terms are assigned probabilities towards each of the LDA-generated topics. Terms with high probabilities towards a particular topic are considered important clues. Terms with very low, or roughly even probabilities among topics are considered unhelpful.

The python module `gensim` implements LDA, generates topics and outputs <supporting_term:factor> tuples. The following are the steps for applying the model:

- Corpus is loaded as a list of strings, one string per document
- Corpus is divided into training and test sets with `sklearn.cross_validation.ShuffleSplit`
- A (TF) vector representation model is created with training data
- The vector representation is converted to an LDA-compatible matrix format
- An LDA model is trained on the matrix-representation of the TF representation of the training data
- The original vector representation model is applied to the test data
- Top topics generated by the LDA model, and top supporting_terms, are used to convert training and test data into feature vectors (see pseudocode outlined in Section 4.3.2)

To match the other methods, we extracted num_topics = [6,36,216] and supporting_term = [10,20]. Since LDA presumes a probability distribution, TF-IDF was not an acceptable input representation and so was not tested. However, LDA accepts two other parameters: `num_passes` and `chunk_size`, see definitions 4.7 and 4.2. Thus, `num_passes` = [1,2,5,10] and `chunk_size` = [100,200,500,1000] were also tested.

4.4 Data

More than just general online content was needed to build a competent ideology detector. Three additional concerns guided the composition of negative data:

1. The need to differentiate peaceful religious language
2. The need to differentiate irrelevant persuasive language
3. The need for contemporaneous language and subject matter; thus a date range that matched the positive corpus (2002-2016)

Given these requirements, it was deemed necessary to look beyond popular but outdated corpora, such as NLTK's built-in Brown Corpus. Manually scraping and cleaning web data was labor intensive; however, it resulted in a corpus tailored to Salafi-jihadist ideology (see Table 4.1). A complimentary date range also ensured that content-detection data could be re-purposed for time-series analysis later on.

Table 4.1. Content Detection Data.

	Datasets	Dates	Samples	Sample Length (Ave)
Positive	Dabiq Magazines	2014-2016	15	26,702
	Rumiyah Magazines	2016	2	22,316
	Inspire Magazines	2010-2015	14	21,848
	Misc Salafi-Jihadist Content	2003-2015	25	4,019
Negative	World News	2014-2016	1732	521
	Blogs	2010-2016	529	454
	Op-Eds	2007-2011	515	624
	Khutbahs (Islamic Sermons)	2000-2016	217	2171

This table shows the distribution of content detection data.

Since visiting extremist websites presents some risk, safe versions of propaganda magazines in PDF were obtained from either ClarionProject.org or Jihadology.net, then converted to plaintext with Adobe Acrobat. Miscellaneous Salafi-jihadist content consisted of speeches, letters and announcements from ideological leaders such as Abu Bakr al'Baghdadi. They were obtained from OpenSource.gov.

Originally, it was thought that splitting magazines into articles would be the most appropriate way to handle the text, since the positive corpus is mostly article-type format. However, this would have excluded a large amount of potentially helpful text, such as captions, forwards and decoratively-displayed text. The resulting disparity in average sample length between the two datasets was handled by normalization.

World News was scraped from *The Atlantic*. Blog posts were obtained from BlogSpot via WebCorp Linguist's Search Engine [40]. Opinions were obtained from *The New York Times*, via their API. Khutbahs were scraped from *KhutbahBank.org*. In all instances of webscraping, `BeautifulSoup`'s hyperlink extraction functionality helped automate crawling and downloading webpage content; however this did not supplant manual cleaning and hypertext removal [41].

4.4.1 Preprocessing

Data was stripped of macrodata (e.g. authors, dates, publisher), html, and article formatting language, such as Extensible Markup Language (XML). Strong cluewords such as “Dabiq” were also added to a stopword list. A comprehensive parsing scheme was applied (see Appendix A) and stopwords were removed (see Appendix B). Several highlights:

- In-term apostrophes were retained, to accommodate Arabic naming conventions
- Significant hyphen handling was employed to mitigate the superfluous word-splitting that typifies multi-column news articles, as well as Arabic multi-term conventions
- Since this thesis focused solely on English text, Arabic unicode characters were stripped. That being said, English-character versions of Arabic terms were retained, e.g. “kuffar” and “hafidhahullah”

4.5 Experiments

Experimentation assumed that the same number of topics would be comparable. Thus, methods were tested across the same size of feature vectors. Experiments are summarized in Table 4.2.

Table 4.2. Content Detection Experiment Summary.

Method	Tested Parameter Values
SRG	not tested (see 4.6.2)

Clustering	vector=[TF,TFIDF]
	K=[6,36,126], supporting_terms=[10,20]
	stemming=[yes,no], lemmatizing=[yes,no]

LDA	num_topics=[6,36,126], supporting_terms=[10,20]
	num_passes=[1,2,5,10], chunk_size=[100,200,500,1000]
	stemming=[yes,no], lemmatizing=[yes,no]

This table summarizes the parameters and methods tested for content detection.

In total, 432 experiments were conducted. In all cases, 10-fold cross-validation was performed and training-test reflected an 80-20 split.

Both stemming and lemmatizing was performed with NLTK implementations (`nltk.stem.porter.PorterStemmer()` and `nltk.stem.wordnet WordNetLemmatizer()`, respectively).

4.6 Results

4.6.1 Scoring

A quick note on how these methods were evaluated: many toolkits contain built-in scoring functions, which typically output an average which values each sample equally. For example, given a million negative documents and twelve positive documents, if a classifier predicted the entire dataset was negative it would still achieve a score of 100%.

This scoring system is not well-suited to an application where the number of positive data is dwarfed by the number of negative data, as is the case with most internet searches. Since this thesis assumes a real-world application would seek to identify a very small portion of data, a binary classification scheme was adopted. In a binary classification, only performance on positive data is reported [42]. Using the same example, a system that correctly classifies the one million negative samples, but none of the twelve positives, would receive a score of 0%.

Although binary scoring makes the results lose some sensationalism, it is a more honest representation of classifier performance.

4.6.2 Semantic Relationship Graph Construction Results

SRG construction was quickly eliminated as the reliable method for discovering concepts. In addition to requiring significant supervision and fine-tuning, it quickly became apparent that WordNet was not well-suited for this corpus. WordNet generated hypernyms for only a small percentage of terms found in the positive corpus, neglecting many of the terms that helped distinguish it as a corpus (see Figure 4.1. Many of the generated hypernyms failed to reflect the usage of the base word, even after implementing a stop word list and removing numeric or single-char terms.

Many readers are probably asking about their obligations towards the Khilafah right now.
'critic.n.02', 'speech_act.n.01', 'written_agreement.n.01', 'tract.n.01

Therefore the Dabiq team wants to convey the position of the Islamic State leadership
on this important matter.
'group.v.02', 'need.n.01', 'bring.v.01', 'state.n.02', 'chemical_phenomenon.n.01',
'ability.n.02', 'content.n.05'

The first priority is to perform hijrah from wherever you are to the Islamic State,
from darul-kufr to darul-Islam.
'position.n.09', 'earliness.n.01', 'act.v.01', 'attribute.n.02', 'civilization.n.01'

Rush to perform it as Musa ('alayhis-salam) rushed to his Lord, saying, And I hastened to
You, my Lord, that You be pleased [Taha: 84].
'move.v.02', 'act.v.01', 'engineering.n.02', 'chemical_element.n.01', 'move.v.02',
'ennoble.v.02', 'speech.n.02'

You can be a major contributor towards the liberation of Makkah, Madinah, and al-Quds.
'containerful.n.01'

Would you not like to reach Judgment Day with these grand deeds in your scales.
'desire.v.01', 'communicate.v.02', 'wisdom.n.03', 'sidereal_time.n.01', 'time_unit.n.01',
'piano.n.01', 'legal_document.n.01', 'metallic_element.n.01', 'quantify.v.02'

Figure 4.1. Typical WordNet Resolutions on Salafi-Jihadist Propaganda.

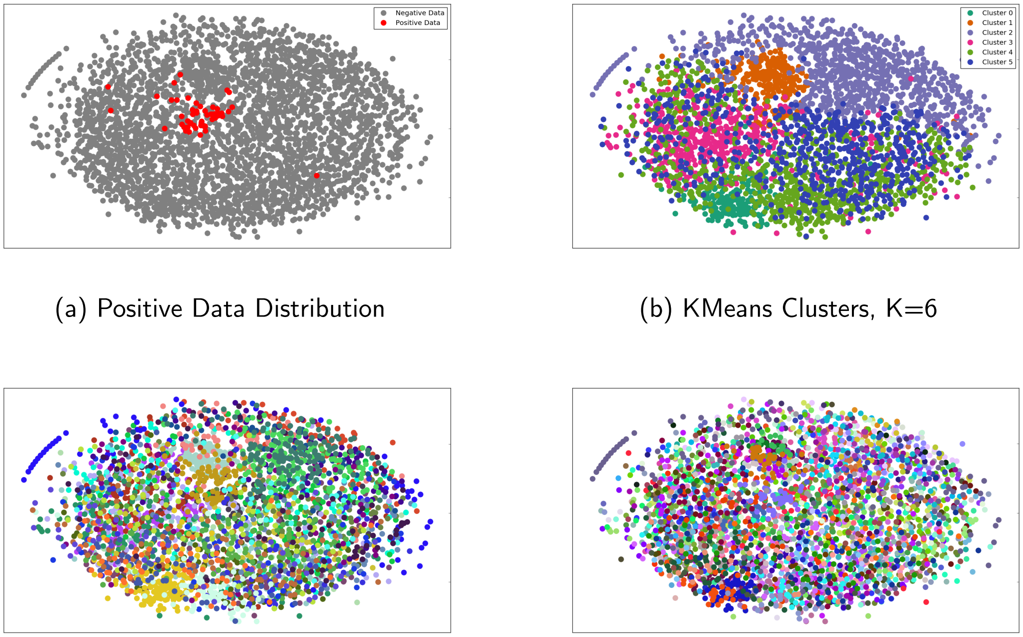
Each textbox contains a sentence from *Dabiq*, the main propaganda magazine of the Islamic State, and its corresponding WordNet resolutions (hypernyms). Hypernyms are in the format of : resolution.part_of_speech.synset. This example is intended to highlight WordNet's inadequacy for this particular dataset and method.

Since WordNet uses the context of a given sentence to choose the best possible synset for a given base word, it was thought that by defining a custom lookup function that references the entire document, the hypernyms would better reflect the context of the document with some minor time-cost. However, quality of generated hypernyms did not improve significantly. As a result, this method was discarded before applying a classifier.

4.6.3 Clustering Results

The subfigures contained in Figure 4.2 show the corpus documents in TF-IDF space. The plots are color-coded either to indicate clusters, or to relate positive (Salafi-jihadist) and negative (non Salafi-jihadist) data.

As a reminder, the objective of clustering was to identify concepts to describe documents for discrimination. Clustering was not intended to discriminate between positive and negative data directly.



(a) Positive Data Distribution

(b) KMeans Clusters, K=6

(c) KMeans Clusters, K=36

(d) KMeans Clusters, K=126

Figure 4.2. Document Clusters in TF-IDF Space.

These figures show document vectors in TF-IDF space. Distances reflect 1 - S_{cos} (see Section 4.3.2). The red and gray image shows positive data (red) amongst negative data (gray). The other three images assign one color to each of the K clusters.

The outlier in subfigure (a) is a 2009 speech made by Abu Omar al-Baghdadi, then leader of the Islamic State. One explanation for the semantic difference may be that he was killed in 2010 and succeeded by Abu Bakr al-Baghdadi before the organization's media arm reached maturity.

Another notable attribute of the cluster plots is the distinctive arch framing the top left portion of the cluster. Upon closer inspection, these were all relatively short texts: all *New York Times* opinion pieces with the exception of one blog.

Best results were achieved with the following parameters:

- stemming = on
- lemmatizing = off
- vector_format = TF
- num_topics = 6
- supporting_terms = 10

This combination achieved scores of 93.57% precision and 66.79% recall (F-score of 76.73%). More results are shown in Table 4.3.

Table 4.3. Clustering Method Results.

vector_format	num_topics	supporting_terms	Precision	Recall	F-Score
TF	6	10	0.93571	0.66795	0.76735
		20	1.00000	0.45390	0.60200
	36	10	0.95417	0.57422	0.70361
		20	0.80000	0.10856	0.18845
	126	10	0.86389	0.30525	0.42668
		20	0.00000	0.00000	0.00000
TF-IDF	6	10	0.05000	0.02000	0.02857
		20	0.74240	0.49510	0.57072
	36	10	0.90714	0.43338	0.55249
		20	0.10000	0.01428	0.02500
	126	10	0.91111	0.46969	0.59186
		20	0.00000	0.00000	0.00000

These results indicate the performance of clustering-derived topics. Best results were achieved with 6 topics, 10 supporting terms and TF vector format.

As seen in Table 4.3, more concepts and more supporting terms resulted in a lower F-score. Generally speaking, doubling the number of supporting terms seemed to result in more dramatic losses than increasing the number of concepts by a factor of six. Increasing the number of supporting terms seemed to produce overlap, making the topics lose some of their power to discriminate. Typical topic-factor pairs generated by clustering are shown in Figure 4.3.

Concept 1 : ‘identified’: 0.00819, ‘suicide’: 0.00470, ‘attacker’: 0.00453, ‘killed’: 0.00342, ‘palestinian’: 0.00240, ‘said’: 0.00192, ‘cox’: 0.01260, ‘third’: 0.00125, ‘yelled’: 0.00159, ‘syrian’: 0.00105

Concept 2 : ‘united’: 0.85052, ‘allies’: 0.00906, ‘member’: 0.00894, ‘israel’: 0.00750, ‘gulf’: 0.00660, ‘government’: 0.00477, ‘european’: 0.00460, ‘would’: 0.00328, ‘states’: 0.00125, ‘russia’: 0.01044

Concept 3 : ‘let’: 1.16708, ‘help’: 0.51209, ‘mercy’: 0.25424, ‘grant’: 0.20875, ‘reminds’: 0.20064, ‘tells’: 0.17134, ‘one’: 0.14514, ‘many’: 0.13972, ‘make’: 0.12991, ‘us’: 0.00205

Figure 4.3. Typical Concepts Generated by Clustering Method.

To demonstrate breadth, three concepts are shown. All were generated with the best performing parameters (with stemming, `vector_format` = TF, `num_topics` = 6, `supporting_terms` = 10. All are abbreviated to five significant figures.

4.6.4 Latent Dirichlet Allocation (LDA) Results

Best results from LDA were obtained with:

- `stemming` = on
- `lemmatizing` = off
- `num_topics` = 126
- `supporting_terms` = 20
- `chunksize` = 100
- `num_passes` = 1

This combination achieved 95.61% precision and 76.53% recall (F-score of 84.38%).

As a reminder, TF format is the only feature vector accepted by LDA. F-scores reflect binary weighting (see section 4.6.1). More results with `num_passes`=1 are enumerated in Appendix C. A typical topic can be seen in Figure 4.4.

‘iran’: 0.05196, ‘deal’: 0.04389, ‘nuclear’: 0.02750, ‘agreement’: 0.02465, ‘unit’: 0.02405, ‘obama’: 0.02315, ‘u’: 0.02096, ‘state’: 0.01944, ‘iranian’: 0.01890, ‘negoti’: 0.01448, ‘sanction’: 0.01441, ‘weapon’: 0.01209, ‘regim’: 0.01206, ‘presid’: 0.01140, ‘iran’: 0.01041, ‘american’: 0.00986, ‘thi’: 0.00971, ‘administr’: 0.00944, ‘kerri’: 0.00920, ‘pragmat’: 0.00869

Figure 4.4. Typical LDA Topic.

Many of the terms are in partial form, e.g. ‘administr’. This is due to stemming, which removes common endings such as ‘ion’. LDA retained out to 18 significant digits; they are displayed here at five, for brevity.

4.7 Conclusion

LDA achieved higher recall than clustering (76.53%, 66.79%, respectively), and so it was chosen as the best content detection method for the combined pipeline outlined in the next chapter. However, the optimal combination of clustering parameters seemed to converge outside of the region of comparison, so further experimentation is recommended. Additionally, it is worth noting that LDA achieved its score with 126 topics comprised of 20 supporting terms, while clustering required only 5 topics comprised of 10 supporting terms.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
The Combined Pipeline: Time Series Detection and Analysis

This chapter seeks to demonstrate how authorship and content detection can be combined to detect propaganda, improve upon the recall of the individual classifiers, and passively monitor an ideology.

5.1 Methodology

After the best method was selected for both authorship and content detection, the two were applied towards time-series analysis as follows.

1. Authorship and content detection classifiers were trained on an initial set of positively and negatively labeled data.
2. Given a new dataset, both classifiers were applied separately. All positive predictions are pooled, to maximize recall.
3. Authorship and content detection classifiers are retrained on old and new data.
4. For each additional dataset, the process is repeated.

For the experiment in topic tracking, the six LDA topics originally generated from the 2004 year group were applied to all other years (2005-2012). For each topic, all samples were translated into feature vectors comprised of a single topic score. A SVM was trained on 2004 data, and tested on the new year group. No re-baselining was performed; that is to say, new data was not integrated into the model, since it would skew performance towards the later year groups (as they would have the most training data).

5.2 Data

To demonstrate the power of this system across time, the negative data from Chapter 4 (Content Detection) was juxtaposed with Islamic State press releases spanning the same date range. See Table 5.1.

These press releases were compiled by Dr. Craig Whiteside and made available by the Naval Postgraduate School's Operations Research Department, namely Dr. Lyn Whitaker and former-student James Friedlein.

Table 5.1. Combined Pipeline Data: Overview.

Label	Data	Dates	Samples	Words per Sample (Ave)
Positive	IS Press Releases	2004-2012	2,811	1,054
Negative	World News	2014-2016	1732	521
	Blogs	2010-2016	529	454
	Op-Eds	2007-2011	515	624
	Khutbahs	2000-2016	217	2171

This table shows the distribution of combined pipeline data.

Data was split into date ranges, one for each year between 2004 and 2012. IS press releases were grouped by publishing date. Negative data was split evenly across these groups due to time constraints, creating the data set shown in Table 5.2.

Table 5.2. Combined Pipeline: Year by Year Corpus.

Label	Data	'04	'05	'06	'07	'08	'09	'10	'11	'12
Positive	IS PR's	83	137	1100	931	190	97	106	80	87
Negative	News	192	192	192	192	192	192	192	192	192
	Blogs	58	58	58	58	58	58	58	58	58
	Op-Eds	57	57	57	57	57	57	57	57	57
	Khutbahs	24	24	24	24	24	24	24	24	24

This table shows data set composition by years, ranging 2004 to 2012.

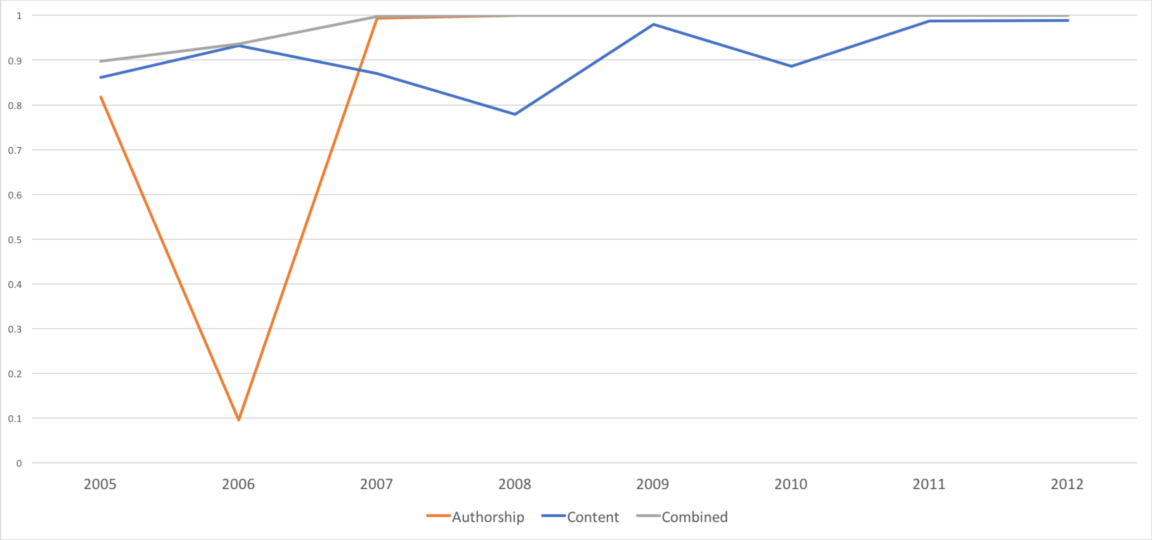
There was a notable surge in positive data for year groups 2006 and 2007. This likely impacted classifier performance during this time range.

5.3 Results

5.3.1 Recall Optimization

When contrasting the two sets of misclassified documents as obtained with the individual classifiers, it became clear how well the classifiers compliment one another. Their differences made them well suited for a system designed to optimize recall, as shown in Figure 5.1.

Figure 5.1. Recall of Authorship and Content Detectors Across Time.

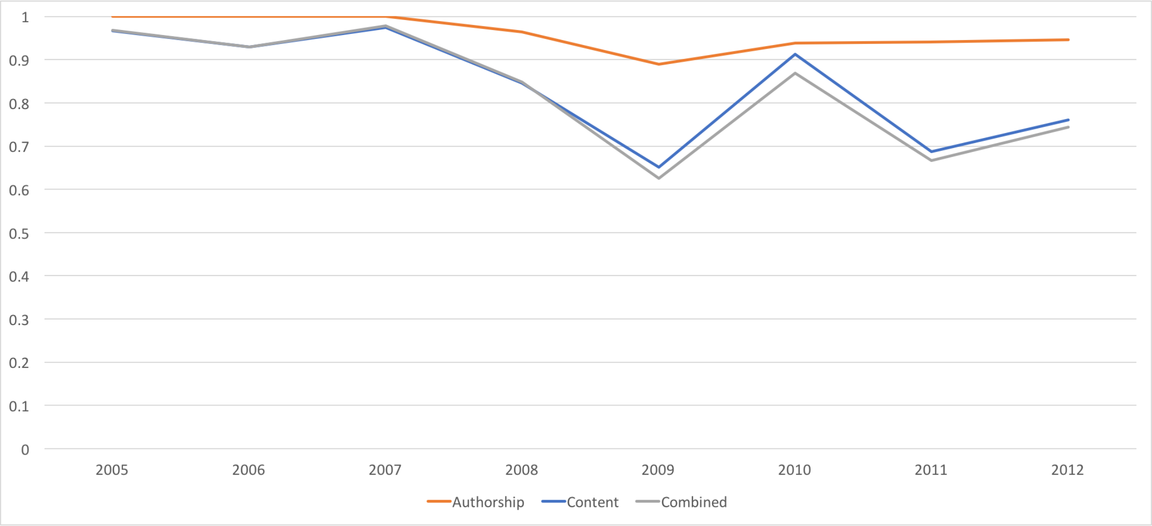


This line graph shows the relative performance of both Authorship and Content Detection (LDA) classifiers across time, in terms of recall.

The initial dip may be attributed to both an imbalance in data, and a semantic mutation. Years 2006 and 2007 had unusually high amounts of positive data. Additionally, there was an imbalance between training and test data: years 2004 and 2005 had fewer samples than 2006 alone (see Table 5.2). Lastly, and likely correlated with the surge in press releases, al Qaeda was headed by Abu Musab al-Zarqawi from late 2004 to mid 2006. The semantic differences illustrated by the content pipeline may represent a sort of semantic footprint of leadership style, or a shift in ideological or organizational messaging.

It is also necessary to mention that pooling positive classifications results in more false positives, which does not affect recall, but does decreases precision (see Figure 5.2).

Figure 5.2. Precision of Authorship and Content Detectors Across Time.



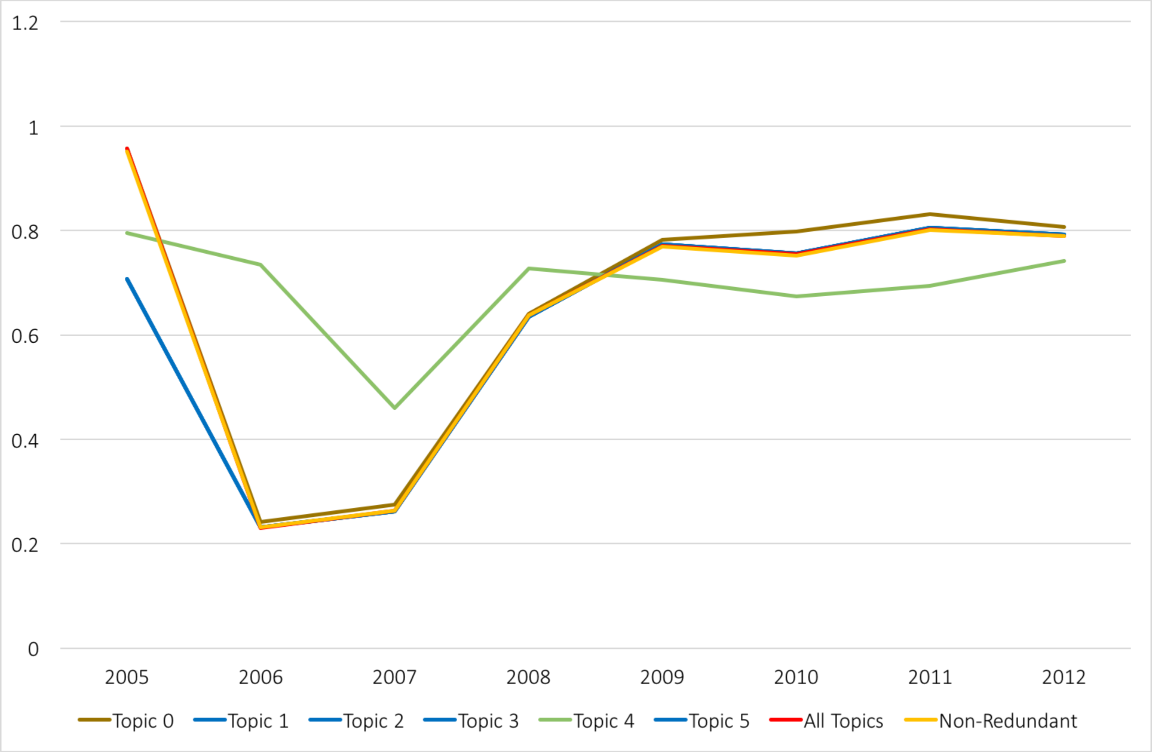
This line graph shows the relative performance of both Authorship and Content Detection (LDA) classifiers across time, in terms of precision. With the pooling of positive labels between both classifiers, some precision is lost.

For information warfare and intelligence applications, it was deemed that an increase in true positives outweighed the cost of decreased precision. Also, with each retraining iteration, the true labels of new data were passed to the classifiers. It is presumed that in a real-world implementation analysts would only see texts classified as positive: so while false positives could be manually rejected, false negatives might never be caught. Therefore a real-world implementation would likely experience lower recall and precision than demonstrated here.

5.3.2 Topic Tracking

When plotted across time in terms of accuracy in Figure 5.3, individual topic performance demonstrated similar trends as the combined performance.

Figure 5.3. Accuracy of Topics Across Time.



This line graph shows the performance of each LDA-generated topic across time, in terms of accuracy. The original six topics generated from the 2004 year group were applied to each new year group. In every test, a support vector machine was trained on 2004 data. Topics 1, 2, 3 and 5 overlapped; they were colored the same to avoid confusion.

Topics 1, 2, 3 and 5 overlapped. Upon closer inspection, they were very similar in supporting-term composition, although their assigned values differed. As a result, predictions based on all six topics closely followed the overlapping topics. Curiously, when redundant topics (2, 3, and 5) were removed from the feature vector, there was no significant improvement in the performance of the feature vector comprised of the remaining two topics. These seems to illustrate that topics are strongly correlated.

5.3.3 Plausible Extensions

Beyond applying this pipeline to new datasets, plausible extensions of a time-series application include:

- Automated re-baselining; that is to say, create new topics to monitor and update machine learning classifiers with new information.
- Tailored scope/detail: this pipeline could be applied on a monthly or weekly basis, depending on the needs of analysts.
- Integrating other data for correlation discovery, e.g. economic data or twitter trend/sentiment analysis, for an added dimension of analysis

5.4 Authorship Validation

For the most part, IS press releases summarized the same metrics with the same terminology. For example, deathtolls in certain areas and results of conflicts. The formulaic nature of these press releases makes it easy to attribute authorship. The question then arose, how well does the authorship classifier perform on ideological content, authored by various individuals and media arms? In other words, can a generic profile of an online ideological author be detected?

To test this, the authorship detector from Chapter 3 was paired with the ideological content data from Chapter 4, rehashed in Table 5.3.

Table 5.3. Content Detection Data.

	Datasets	Dates	Samples	Sample Length (Ave)
Positive	Dabiq Magazines	2014-2016	15	26,702
	Rumiyah Magazines	2016	2	22,316
	Inspire Magazines	2010-2015	14	21,848
	Misc Salafi-Jihadist Content	2003-2015	25	4,019
Negative	World News	2014-2016	1732	521
	Blogs	2010-2016	529	454
	Op-Eds	2007-2011	515	624
	Khutbahs (Islamic Sermons)	2000-2016	217	2171

This table shows the distribution of content detection data.

With the optimal SVM parameters determined in Chapter 3 (sigmoid kernel, C=1, gamma=.0001), Salafi-jihadist material was detected with 99.84% accuracy and 95.42%

recall with an equally-weighted (vice binary-weighted) scoring scheme. In other words, a generic profile of an online Salafi-jihadist was differentiated from a large range of online media, with greater accuracy than the editor of a relatively small section of a News outlet was, from among his own employees. This is reasonable, considering how difficult it is to differentiate an editor, from writers he edits, all covering the same subject matter and writing for the same website. However, it is still an achievement to attain such high accuracy for something relatively generic.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6:
Conclusion and Future Work

“Ultimately, it is not going to be enough to defeat ISIL in the battlefield. We have to prevent it from radicalizing, recruiting and inspiring others to violence in the first place. And this means defeating their ideology.” -President Barack Obama, to the United Nations General Assembly [29]

6.1 Conclusions

Our results show that ideology is detectable in text. It goes further to demonstrate that by pairing a ideological-content detector with an authorship detector, higher recall can be obtained pursuant to intelligence gathering applications. The same techniques employed to extract the underlying semantic relationships that define an ideology, were also shown to be measurable across time, and thus able to be monitored.

Another notable contribution of this thesis, is the data itself: 3,049 samples (2,787,579 words) of modern web content were collected and cleaned. As a whole, this corpus was tailored for studying Salafi-jihadism as manifested in text.

This research directly benefits modern counter propaganda and disinformation operations. It also has profound implications for future applications, including automatically generating counter propaganda content, or retailoring this system to other ideologies. That being said, in its current state, this work has two major weaknesses: it is overly specialized for analyzing Salafi-jihadist propaganda, and it comes with considerable manpower requirements. The amount of time and effort to oversee, scrape and clean data from the internet is extensive and should not be overlooked. However, this is something that could be expedited with applications, for a cost.

Another consideration, is the inherent risk associated with downloading large amounts of materials from illegitimate websites. Malware was accidentally downloaded at least twice during research and data collection. It is recommended that such a system would only be implemented on an isolated, expendable network.

6.2 Recommendations for Future Work

For an ideology-agnostic system, or one with better recall, more work is needed. It is recommended that the following areas are explored for a more robust detection scheme.

6.2.1 Explore the Notion of "Embeddedness"

One writing element that Salafi-jihadist leaders leverage particularly well, is voice. It conveys education, experience and influences; all of which are critical to relating to someone. Things an author might try to convey include an Oxford education, a humble upbringing, or heavy religious influences. Piety is especially powerful, because it conflates the author's arguments with morality. In the early stages of this thesis, we experimented with the notion of "embeddedness," or the way ideological proponents embed religion in their arguments. For example, Salafi-jihadist leaders typically open with a short prayer, and saturate their writings with quotes from the Koran and the Hadith. Religious extremists also tend to prefer certain religious excerpts over others. Thus, it should be possible to build detectors for radical usage, particular to a religion. Other things that could become part of a feature vector include enclosures, such as apostrophes and brackets, and Arabic phrases.

6.2.2 Integrate Other Media

To counter increasingly professional and prolific media arms, a robust detection scheme could benefit from incorporating other communication mediums. Images and video are particularly powerful tools for persuasion, since they are well-suited for eliciting emotion and conveying symbolism. Complimenting text analysis with object detection in images could yield higher accuracy or new knowledge. Sourcing propaganda from the dark web, or incorporating other metrics, such as stock market data, could also lead to interesting correlations.

6.2.3 Try Additional Technique Variations

Some thoughts for varying techniques outlined here include trying more clustering algorithms and experimenting with trigrams (or longer n-grams) for supporting-term creation. Optimizing the SVM parameters for ideology detection may also yield improved results.

6.2.4 Experiment with New Techniques

In addition to variations of techniques laid out here, other techniques worth exploring include Latent Semantic Analysis (LSA), Non-negative Matrix Factorization (NMF), and neural networks. Sci-kit learn has modules for implementing LSA and NMF for topic creation, and they could be easily contrasted with the clustering and Latent Dirichlet Allocation methods outlined here. Neural Networks, on the other hand, may represent the next step in application. In the future, generative modeling may be the means of both generating and countering propaganda [43]. Generative modeling may also signal the ultimate mastering of communication, since it requires proficiency in grammar, composition and persuasion. Neural networks may be the best method for creating such a model. Recurrent Neural Networks (RNNs) are particularly adept at handling time-series data and have been used to generate music. Long Short Term Memory (LSTM) networks, a subset of RNNs, show promise for generating text, since they default to integrating new info with learning long-term dependencies [44]. This may mean that they could proficiently fit new information (such as a given vocab, event or idea) into long-term patterns (e.g. grammar rules) to produce convincing text. Auto-generation of counter propaganda could advantage Information Operations in terms of product-volume, and conservation of human resources.

6.2.5 Optimize for Time-Savings

The major weakness of this pipeline is the time required to find, extract and clean online texts. A system worthy of implementation would need to be better optimized for time-savings. This would likely require a demonstration of automated web crawling, as well as automated ideology monitoring. This might include a visualization-heavy user interface, and the integration of an alert system.

6.2.6 Experiment with Other Ideologies

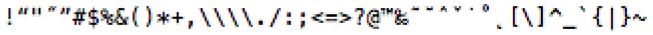
Many things we did to improve performance on Salafi-jihadism, made this work less applicable to other ideologies. For example, in-word apostrophes and hyphens were retained, to accommodate common conjugations (e.g. *al-Qa'ida* is a common variant of *al Qaeda*). Expanding this work to detect other ideologies could lend new insight to natural language processing and propaganda, in addition to broadening intelligence capabilities. Thoughts include studying propaganda from the People's Republic of China, which likely employs different techniques for internal population control, and foreign manipulation. Another fun experiment would be to contrast statements from two opposing politicians, such as election candidates, or established administrations. An ideological comparison of statements from the Kremlin and White House over time could produce interesting results.

APPENDIX A:
Content Parsing Method

Method for parsing for content-detection follows:

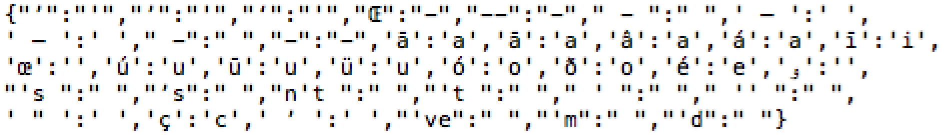
1. Read in file, as a single string
2. Replace `sundesired_punctuation` with white space
3. Replace `special_replace` items with a mapped value
4. Remove arabic lettering
5. Lowercase all text, split string on white spaces (doc is now in list format)
6. Lemmatize, if applicable
7. Stem, if applicable
8. Remove stopwords (see Appendix B)
9. Remove numbers

`undesired_punctuation` items:



```
!“"”"#$%&()*+,\\\\./:;<=>?@™&¨˘^ˇ·°‚[\]^_`{|}~
```

`special_replace` items (in the format of <item:replacement value>):



```
{"’":"'","‘":"'","′":"'","Œ":"–","––":"–"," – ":" ",' – ':' ',
' – ':' '," –":" ","–":"–",'ā':'a','ā':'a','â':'a','á':'a','ī':'i',
'œ':'','ú':'u','ū':'u','ü':'u','ó':'o','ǒ':'o','é':'e','ˌ':'',
"'s ":" ","'s":" ","n't ":" ","'t ":" "," ' ":" "," '' ":" ",
' " ':' ','ç':'c',' ’ ':' ',"'ve":" ","'m":" ","'d":" "}
```

Arabic was defined as `[chr(n) for n in range(`0x600, 0x700`)]`

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B:
Stopwords

The following are the terms were removed during pre-processing:

corpus_stopwords = [’http’,’com’,’org’,’net’,’url’,’www’,’khutbah’,’khutbahs’, ’khutbah-bank’,’theatlantic’,’atlantic’,’newyorktimes’,’dabiq’,’inspire’,’october’, ’rumiyah’]

date_info = [’January’,’February’,’March’,’April’,’May’,’June’,’July’,’August’,’September’, ’October’,’November’,’December’,’Jan’,’Feb’,’Mar’,’Apr’,’Jun’,’Jul’,’Aug’,’Sep’, ’Oct’,’Nov’, ’Dec’,’2001’,’2002’,’2003’,’2004’,’2005’,’2006’,’2007’,’2008’,’2009’,’2010’, ’2011’,’2012’, ’2013’,’2014’,’2015’,’2016’,’2017’]

NLTK’s standard 153-term English stopword list (from nltk.corpus.stopwords.words(’english’)) = [’i’, ’me’, ’my’, ’myself’, ’we’, ’our’, ’ours’, ’ourselves’, ’you’, ’your’, ’yours’, ’yourself’, ’yourselves’, ’he’, ’him’, ’his’, ’himself’, ’she’, ’her’, ’hers’, ’herself’, ’it’, ’its’, ’itself’, ’they’, ’them’, ’their’, ’theirs’, ’themselves’, ’what’, ’which’, ’who’, ’whom’, ’this’, ’that’, ’these’, ’those’, ’am’, ’is’, ’are’, ’was’, ’were’, ’be’, ’been’, ’being’, ’have’, ’has’, ’had’, ’having’, ’do’, ’does’, ’did’, ’doing’, ’a’, ’an’, ’the’, ’and’, ’but’, ’if’, ’or’, ’because’, ’as’, ’until’, ’while’, ’of’, ’at’, ’by’, ’for’, ’with’, ’about’, ’against’, ’between’, ’into’, ’through’, ’during’, ’before’, ’after’, ’above’, ’below’, ’to’, ’from’, ’up’, ’down’, ’in’, ’out’, ’on’, ’off’, ’over’, ’under’, ’again’, ’further’, ’then’, ’once’, ’here’, ’there’, ’when’, ’where’, ’why’, ’how’, ’all’, ’any’, ’both’, ’each’, ’few’, ’more’, ’most’, ’other’, ’some’, ’such’, ’no’, ’nor’, ’not’, ’only’, ’own’, ’same’, ’so’, ’than’, ’too’, ’very’, ’s’, ’t’, ’can’, ’will’, ’just’, ’don’, ’should’, ’now’, ’d’, ’ll’, ’m’, ’o’, ’re’, ’ve’, ’y’, ’ain’, ’aren’, ’couldn’, ’didn’, ’doesn’, ’hadn’, ’hasn’, ’haven’, ’isn’, ’ma’, ’mightn’, ’mustn’, ’needn’, ’shan’, ’shouldn’, ’wasn’, ’weren’, ’won’, ’wouldn’]

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C:
Latent Dirichlet Allocation (LDA) Results

The following are results from the best combination of preprocessing parameters (with stemming, without lemmatizing), and `num_passes=1`.

Table C.1. LDA Results: `num_passes=1`.

`chunksize`	`num_topics`	`supporting_terms`	Precision	Recall	F-Score
100	6	10	0.94750	0.60098	0.71985
		20	0.93737	0.72582	0.80555
	36	10	0.97738	0.70995	0.81142
		20	0.94439	0.76335	0.83920
	126	10	0.96314	0.74793	0.83473
		20	0.95609	0.76529	0.84377
200	6	10	0.97750	0.62098	0.74262
		20	0.93646	0.66941	0.76228
	36	10	0.96591	0.65766	0.76696
		20	0.97146	0.72514	0.82183
	126	10	0.97981	0.74070	0.83731
		20	0.95424	0.75757	0.83685
300	6	10	0.95750	0.59473	0.71729
		20	0.96071	0.69313	0.79436
	36	10	0.97917	0.69129	0.80051
		20	0.96174	0.69167	0.79450
	126	10	0.97007	0.73801	0.83239
		20	0.97738	0.71892	0.82157

chunksize	num_topics	supporting_terms	Precision	Recall	F-Score
400	6	10	0.97750	0.62098	0.74262
		20	0.95265	0.68190	0.78393
	36	10	0.96778	0.67518	0.78556
		20	0.96841	0.68270	0.79151
	126	10	0.95822	0.66988	0.77116
		20	0.95757	0.67916	0.78447
500	6	10	0.97750	0.58705	0.72055
		20	0.96841	0.64690	0.76623
	36	10	0.97571	0.67599	0.78459
		20	0.97841	0.69065	0.79984
	126	10	0.96551	0.65322	0.77401
		20	0.97738	0.68592	0.79991

List of References

[1] E. Fink, J. Pagliery, and L. Segall. (2015, Nov. 24). Technology and the fight against terrorism. CNN Money. [Online]. Available: http://money.cnn.com/2015/11/24/technology/targeting-terror-intelligence-isis/

[2] C. Herridge. (2015, May 26). Bulletin warns US analysts overwhelmed by pro-ISIS social media, military posts threatened. [Online]. Available: http://www.foxnews.com/politics/2015/05/26/bulletin-warns-us-analysts-overwhelmed-by-pro-isis-social-media-military-posts.html

[3] J. Moore. (2015, Feb. 35). Western intelligence services overwhelmed by jihadist propaganda. newsweek.com. [Online]. Available: http://www.newsweek.com/2015/03/06/western-intelligence-services-overwhelmed-jihadist-propaganda-309263.html

[4] The Data Team. (2016, Aug.12). The evolution of the Islamic State. The Economist. [Online]. Available: http://www.economist.com/blogs/graphicdetail/2016/08/daily-chart-10

[5] L. Loveluck. (2015, Jun. 8). Islamic State: Where do its fighters come from? The Telegraph. [Online]. Available: http://www.telegraph.co.uk/news/worldnews/islamic-state/11660487/Islamic-State-one-year-on-Where-do-its-fighters-come-from.html

[6] D. Harris. (2014, Sep. 9). The Islamic State's (ISIS, ISIL) magazine. The Clarion Project. [Online]. Available: http://www.clarionproject.org/news/islamic-state-isis-isil-propaganda-magazine-dabiq. Accessed: Oct 2016.

[7] Introduction to Data Science. (2016). University of Utah. [Online]. Available: http://datasciencecourse.net/2016/

[8] R. Schutt and C. O'Neil, *Doing data science: Straight talk from the frontline*. O'Reilly Media, Inc., 2013.

[9] J. Han and M. Kamber, *Data Mining Concepts and Techniques*. Morgan Kaufmann, 2006.

[10] J. Friedlein, "The Islamic State Battle Plan: Press Release Natural Language Processing," Master's thesis, Naval Postgraduate School, Monterey, California, 2016.

[11] Y. Wang, "Various approaches in text pre-processing," *TM Work Paper No*, vol. 2, no. 5, 2004.

[12] Alphabetical list of part-of-speech tags used in the Penn Treebank Project. (2003). University of Pennsylvania. [Online]. Available: https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[13] N. Y. Habash, *Introduction to Arabic Natural Language Processing*. Morgan & Claypool, 2010.

[14] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Education, Inc., 2009.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[16] T. Pang-Ning, M. Steinbach, V. Kumar *et al.*, "Introduction to data mining," in *Library of congress*, 2006, vol. 74.

[17] H. Sawaf, J. Zaplo, and H. Ney, "Statistical classification methods for arabic news articles," *Natural Language Processing in ACL2001, Toulouse, France*, 2001.

[18] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[19] T. Gonçalves and P. Quaresma, "Evaluating preprocessing techniques in a text classification problem," *São Leopoldo, RS, Brasil: SBC-Sociedade Brasileira de Computação*, 2005.

[20] E. Alpaydin, *Introduction to machine learning*. MIT press, 2014.

[21] C. N. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts." in *COLING*, 2014, pp. 69–78.

[22] S. Reid, "10 misconceptions about neural networks," May 2014. Available: http://www.turingfinance.com/misconceptions-about-neural-networks/

[23] G. Jowett and V. O'Donnell, *Propaganda and Persuasion*. Sage Publications, Inc., 1999.

[24] A. Moghadam, "The salafi-jihad as a religious ideology," *The Combating Terrorism Center Sentienel at West Point*, vol. 1, no. 3, pp. 1–3, 2008.

[25] Frontline, "Analysis: Wahhabism," 2014. Available: http://www.pbs.org/wgbh/pages/frontline/shows/saudi/analyses/wahhabism.html

[26] F. A. Gerges. (2016, Mar. 18). The World According to ISIS. Foreign Policy Journal. [Online]. Available: https://www.foreignpolicyjournal.com/2016/03/18/the-world-according-to-isis/

[27] U.S. House. 114th Congress, 1st Session. (2016, May). *H.R.5181, Countering Foreign Propaganda and Disinformation Act*. [Online]. Available: https://www.congress.gov/bill/114th-congress/house-bill/5181/all-info

[28] C. Timberg. (2016, Nov. 30). Effort to combat foreign propaganda advances in Congress. The Washington Post. [Online]. Available: https://www.washingtonpost.com/business/economy/effort-to-combat-foreign-propaganda-advances-in-congress/2016/11/30/9147e1ac-e221-47be-ab92-9f2f7e69d452_story.html?utm_term=.73923e33c438

[29] G. Miller and K. DeYoung. (2016, Jan. 8). Obama administration plans shake-up in propaganda war against ISIS. The Washington Post. [Online]. Available: https://www.washingtonpost.com/world/national-security/obama-administration-plans-shake-up-in-propaganda-war-against-the-islamic-state/2016/01/08/d482255c-b585-11e5-a842-0feb51d1d124_story.html?utm_term=.97fee7127263

[30] J. Arquilla and D. A. Borer, *Information strategy and warfare: a guide to theory and practice*. Routledge, 2007.

[31] J. Hudson. (2013, May 3). How jihadists schedule terrorist attacks. Foreign Policy. [Online]. Available: http://foreignpolicy.com/2013/05/03/how-jihadists-schedule-terrorist-attacks/

[32] A. Kott, *Information Warfare and Organizational Decision-making*. {Artech House Publishers}, 2006.

[33] H. Love, *Attributing authorship: an introduction*. Cambridge University Press, 2002.

[34] C. Cortes and V. Vapnik, “Support-vector networks,” no. 3. Springer, 1995, vol. 20, pp. 273–297.

[35] B. Gelfand, M. Wulfekuler, and W. Punch, “Automated concept extraction from plain text,” in *AAAI 1998 Workshop on Text Categorization*, 1998, pp. 13–17.

[36] P. Buitelaar, P. Cimiano, and B. Magnini, *Ontology learning from text: methods, evaluation and applications*. IOS press, 2005, vol. 123.

[37] Princeton University. (2015). WordNet: a lexical database for English. [Online]. Available: https://wordnet.princeton.edu/

[38] I. S. Dhillon, “Co-clustering documents and words using bipartite spectral graph partitioning,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 269–274.

[39] B. Rose. (2016). Document clustering with python. [Online]. Available: http://brandonrose.org/clustering

[40] “Webcorp linguist’s search engine,” http://wse1.webcorp.org.uk/home/index.html, accessed: 2016-011-30.

[41] “Beautiful soup documentation,” https://www.crummy.com/software/BeautifulSoup/bs4/doc/, accessed: 2016-011-30.

[42] scikit learn. (2016). sklearn.metrics.f1_score documentation. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

[43] P. Soulos. (2017, Mar. 7). Machine learning and misinformation. paulsoulos.com. [Online]. Available: http://paulsoulos.com/editorial/2017/03/07/machine-learning-and-misinformation.html

[44] C. Olah. (2015, Aug. 27). Understanding LSTM networks. colah’s blog. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California